

Universidad Carlos III de Madrid



Proyecto de fin de carrera

Filtro paso bajo discreto de operación directa sobre señales PWM

Autor: Sergio López Batres

Tutor: Luis Hernández Corporales

Titulación: I.T.I Electrónica Industrial



Agradecimientos:

Para todos los que me han apoyado durante la carrera, especialmente a mis padres, por meterme desde pequeño la idea de que estudiar merece la pena y que no dejara nunca de seguir adelante, que intentara ser el mejor, también a mi hermano por ayudarme durante todo este tiempo en cualquier duda que tenía y en darme ánimos simplemente con su ejemplo, una de mis mayores motivaciones fue ver en lo que él había conseguido con su esfuerzo. Parte de estos agradecimientos deben ir para mi abuela, ya que su orgullo por ver mi éxito en mis estudios es también mi orgullo. También se lo dedico a los que no pueden ver cómo he alcanzado mi meta porque ya no están aquí, pero estoy seguro que lo están viendo desde arriba y están tremendamente felices por mí (va por ti Pepe). Por último no podría olvidar a mi tutor Luis por todas las tutorías que me ha dado y el tiempo que me ha dedicado, sin sus grandes ideas y sin su ayuda habría sido imposible que esto saliera adelante. Aunque para finalizar no me olvido de mis compañeros, por todas esas tardes de prácticas, por todos esos duros momentos de exámenes y esos mensajes y llamadas de felicidad cuando aprobábamos asignaturas difíciles (¡espero que nos veamos en la superior!).

Descripción del proyecto:

Este proyecto aborda la construcción y diseño de un filtro de paso bajo que opera sobre señales moduladas en anchura de impulsos (PWM). Utiliza componentes analógicos y un microprocesador que efectúa el control del sistema. Su función es servir de atenuador de los armónicos en amplificadores de clase D que podrían usarse en sistemas de audio, sin necesidad de un filtro paso bajo analógico de orden elevado.

Índice

1	5
1.- Introducción.	5
2	
2.1.- La célula básica del circuito, la célula de retraso.	14
2.2.- Ejemplo de filtro de 1 ^{er} orden.	20
2.3.- Diagrama de bloques del demostrador.	23
2.3.1.- Diagrama de bloques del demostrador simulado.	24
2.3.2.- Diagrama de bloques del circuito construido.	25
2.4.- Elección de componentes semiconductores y microprocesador.	28
3	
3.- Diseño de hardware.	32
3.1.- Esquemas eléctricos.	33
3.1.1.- Esquema eléctrico de simulación del circuito.	33
3.1.2.- Esquema eléctrico real del circuito para su fabricación (Figura 30)	36
3.2.- Simulaciones.	41
4	
4.- Diseño de Software.	51
4.1.- Organigrama del programa.	52
4.1.1.- Organigrama del programa principal.	52
4.1.2.- Organigrama de la interrupción del A/D.	54
4.1.3.- Organigrama de las interrupciones externas 0 y 1.	59
	5.- Construcción y pruebas.
	5.1.- Diseño de la PCB.
	5.1.1.- Huellas de los componentes.
	5.1.2.- Ancho de las pistas.
	5.1.3.- Pads de los componentes.
	5.1.4.- Capas de la PCB.
	5.1.5.- Plano de masa.
	5.2.- Montaje.
	5.3.- Pruebas.
	5.3.1.- Listado de pruebas a realizar.
	5.3.2.- Señales de control.
	5.3.3.- Funcionamiento como célula de retraso.
	5.3.4.- Funcionamiento como filtro de primer orden.
	6
	6.- Conclusión.
	7
	7.- Bibliografía.
	8
	8.- Presupuesto
	9
	9.- Anexos.
	9.1.- Listado de programa.
	9.1.1.- Código fuente del programa original.
	9.1.2.- Código fuente del programa para probar el circuito en laboratorio.
	9.1.3.- Código fuente del programa para probar la célula de retraso.
	9.2.- Hojas de características. (Datasheet) ...



1.- Introducción.



La escena de la conversión de datos ha cambiado drásticamente en estos últimos años. Mientras que las arquitecturas clásicas como el pipeline, SAR y los conversores sigma delta de interruptores capacitivos han ganado su propia posición en las tecnologías CMOS por debajo del micrómetro, la nueva escala nanométrica en las tecnologías requiere un marco de trabajo completamente nuevo. Esto es debido principalmente a la imposibilidad de adecuación de la tecnología de transistores MOS en los circuitos analógicos clásicos.

Por otro lado, los pilares de la electrónica analógica de alto rendimiento requerida por la industria del multimedia y las comunicaciones han sido comúnmente los amplificadores, los DACs y los relojes de baja fluctuación. Los amplificadores logran linealidad a través de la realimentación y la alta ganancia mientras que los DACs logran la linealidad a través de su layout y técnicas de mejora digitales.

Sin embargo, ninguno de estos dispositivos está realmente obligado a adquirir y procesar señales analógicas con alta resolución. Por ejemplo, el cerebro de los animales es capaz de percibir señales de audio y vídeo con un incomparable rendimiento sin tener dispositivos lineales como amplificadores operacionales o referencias de tiempo precisas como las que se podrían conseguir con los cristales de cuarzo.

Estos hechos han orientado la investigación hacia las arquitecturas de conversión de datos, donde el número de transistores en la región activa es minimizado. Para lograr esta meta, están prosperando diferentes topologías donde la señal es codificada tanto en dominio del tiempo como en dominio de la amplitud. La ventaja de estos sistemas reside en el uso de señales binarias codificadas en el tiempo que no requieren dispositivos electrónicos de alto rendimiento. Las neuronas son un ejemplo de sistemas biológicos que usan un tipo de codificación similar.

La representación y procesamiento de señales analógicas con lógica digital asíncrona ha sido probada con éxito y obteniendo buenos resultados. Sin embargo, estos circuitos no están basados en principios uniformes de velocidad de muestreo, lo que complica su aplicación práctica.

Este proyecto presenta una posible solución justamente para esto que hemos comentado sobre la codificación y el procesamiento de señales muestreadas uniformemente como por ejemplo un circuito de interruptores capacitivos haría, pero sin la necesidad de amplificadores operacionales de respuesta rápida y alta precisión y otros dispositivos de alto rendimiento, pudiéndole dar varios usos, aunque el uso en el que nos centraremos será en el de filtro de paso bajo de 1^{er} orden por su simplicidad

Aparte de su aplicación teórica en la codificación de ondas, otro tema del que sería muy interesante hablar es como se aplicaría en sistemas de amplificación, los amplificadores de clase D han mejorado la eficiencia del audio y de los amplificadores de potencia xDLS en los últimos años. Estos amplificadores tipo D tienen una gran calidad y son capaces de mejorar de una forma muy significativa el rendimiento y la calidad de las señales en los circuitos en los que se aplican.

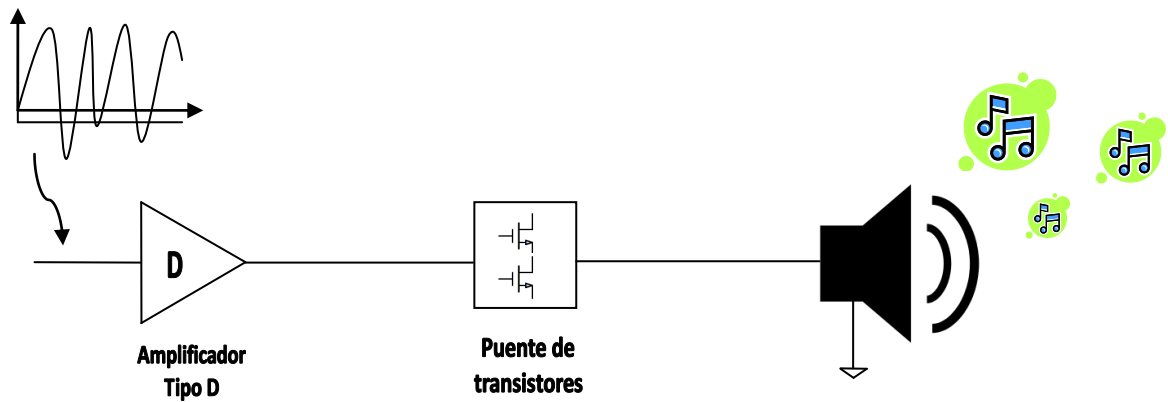


Figura 1: Posible aplicación de un amplificador tipo D en sistema de audio.

En las dos aplicaciones anteriormente comentadas la señal analógica viene de una fuente digital muestreada, que es entonces convertida en una señal analógica mediante un convertidor digital analógico, y seguida por un modulador de ancho de pulso analógico (PWM) que controla al amplificador tipo D. Como podemos observar en el esquema de la figura 2.

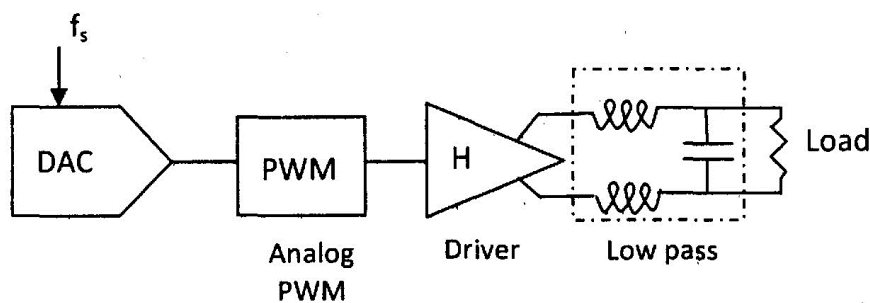


Figura 2: Un amplificador de potencia clase D controlado por un DAC.

En esta solución, el sistema requiere integrar un convertidor digital analógico multibit de alta linealidad junto con la etapa de control de potencia, lo que sin duda da una cierta complejidad al mismo. En cambio, la figura 3 muestra una configuración alternativa, donde una señal Sigma-Delta modulada de un único bit es usada para manejar directamente la etapa de potencia. Esta señal modulada es una señal binaria igual que la salida del modulador PWM de la figura 2, lo que nos ahorraría así un bloque de modulación de señales PWM obteniendo un sistema más sencillo.

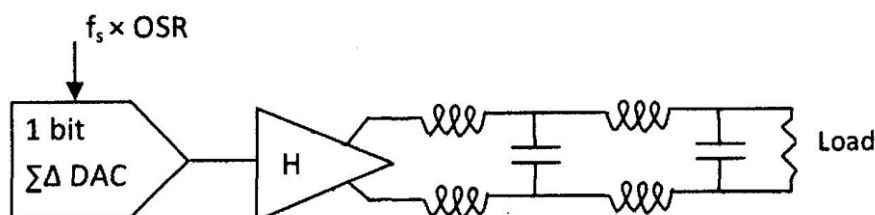


Figura 3: Un modulador sigma delta controlando una carga de potencia.

El problema que tiene esta solución sin embargo, es que el alto ratio de sobre muestreo requerido por un convertidor digital analógico Sigma-Delta de un único bit y la cantidad de ruido de cuantificación fuera de banda, fuerza a usar un filtro paso bajo de alto orden tras la etapa de potencia. Este filtro de salida requiere una gran cantidad de bobinas y condensadores discretos fuera del chip, haciendo esta opción menos atractiva en un principio que la solución dada por la figura 2.

Este proyecto de fin de carrera mostraría una solución innovadora para solucionar el problema de la figura 3 ya que realiza un filtrado paso bajo directamente en los pulsos de la señal Sigma-Delta. El circuito propuesto proporciona un suavizado en el dominio del tiempo de las transiciones del ancho de pulso que atenúan el ruido de cuantificación presente en una señal de un único bit, haciendo que de esta forma el filtro de paso bajo pasivo requerido a la salida del controlador de potencia pudiera ser muy simplificado. Esta solución mantiene la simplicidad y la linealidad de un convertidor analógico digital Sigma-Delta de un único bit y permite la implementación de todo el sistema en una tecnología con bajo voltaje de alimentación. Así el sistema puede ser implementado en un chip integrando hardware digital de alta densidad junto con etapas amplificadoras de potencia, sin recurrir a complicada circuitería fuera del propio chip para cumplir con las restricciones de EMC y niveles de señal fuera de la banda apropiada.

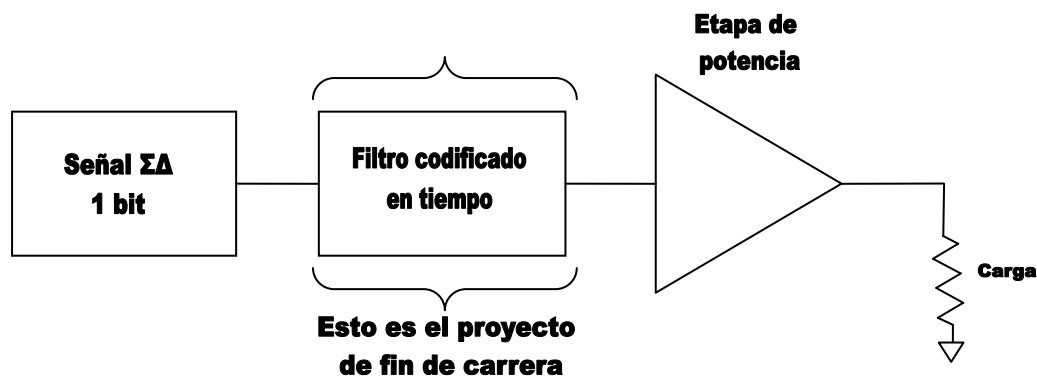


Figura 4: Solución alternativa para realizar el filtrado paso bajo a los pulsos de la señal Sigma-Delta.



2.- Diseño del sistema.

Antes de comenzar a ver cómo está diseñado a nivel de circuito todo el sistema, primero hay que saber qué tipo de señales va a procesar, puesto que es la base de su diseño y funcionamiento. Como se puede ver en la introducción, desde un principio se habla sobre señales codificadas en el tiempo, señales binarias etc. Que no son otra cosa que señales PWM (Modulación de ancho de pulso) y que serán las que entrarán al filtro definitivamente, puesto que este tipo de modulación es el más común a la hora de codificar señales continuas o codificadas en amplitud en señales codificadas temporalmente.

La modulación de señales en tiempo continuo en señales PWM se puede hacer de una forma muy sencilla y sin la necesidad de añadir al circuito una gran cantidad de componentes extra. Para ello primero se necesitaría un circuito que tomara una cantidad de muestras suficientes de la señal original (circuito Sample & Holder) para que luego, tras modificarla, codificarla etc. Ésta se pudiera reconstruir. Según el teorema de Nyquist para el muestreo de señales, la frecuencia mínima a la que se han de tomar las muestras para recuperar una señal correctamente, es decir la frecuencia de muestreo, ha de ser como mínimo el doble que la frecuencia de la señal a muestrear ($f_m = 2 \cdot f_t$), si se muestrearan las señales a una frecuencia menor a esta, no se aseguraría en ningún momento que la señal recuperada sea igual que la señal introducida en el modulador. Una vez muestreada la señal en tiempo continuo a la entrada, el siguiente paso es generar la señal modulada en tiempo, para ello simplemente se necesita un comparador y una onda de diente de sierra o triangular para así ser capaces de obtener una salida de pulsos de distinto ancho, tras unir todos los conceptos se obtendría un circuito similar al siguiente.

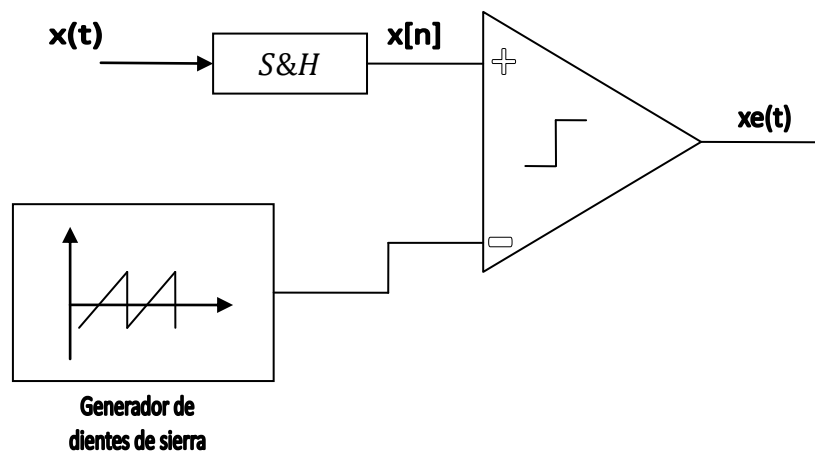


Figura 5: Codificador de ondas PWM.

Siendo $x(t)$, la señal de entrada continua al circuito, $x[n]$ las muestras de la señal de entrada realizadas por el muestreador (S&H), y por último $xe(t)$ la señal de salida codificada en tiempo (PWM).

El funcionamiento de este circuito es muy sencillo, ya que el ancho de pulso a la salida del comparador dependerá directamente de la amplitud de la muestra tomada, si la muestra es de una amplitud muy elevada, entonces el ancho del pulso de salida será elevado, mientras que si la muestra es de una amplitud pequeña, el pulso de salida será muy fino. Una forma de explicarlo más detalladamente sería decir que el ancho de pulso

de salida depende directamente de cuánto tiempo sea superior el valor de la tensión de la señal en la patilla positiva del comparador hasta que le alcance el valor de la señal en patilla negativa, puesto que la salida del comparador estará a nivel alto mientras la tensión en su entrada positiva sea mayor a la tensión en su entrada negativa, en cuanto la tensión de esta última entrada supere a la de la primera, la salida del comparador permutará a nivel bajo.

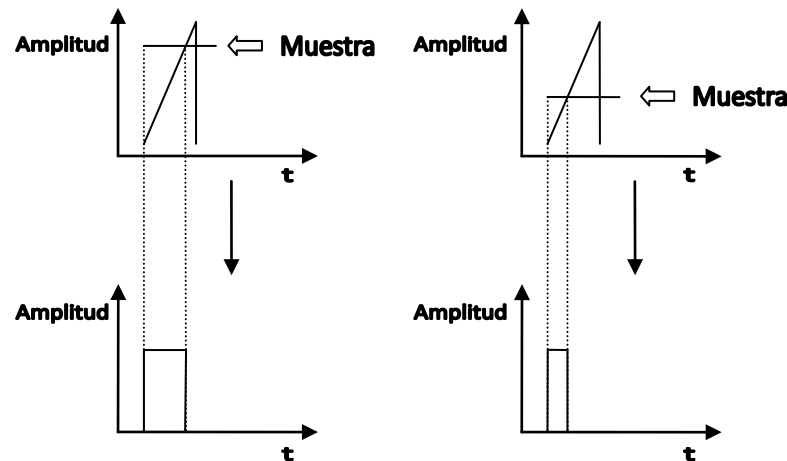


Figura 6: Funcionamiento del modulador de ondas PWM.

En la figura 6 se puede ver perfectamente lo explicado, la muestra de la izquierda es de una tensión más elevada y por tanto el ancho de pulso a la salida del comparador está más tiempo a nivel alto que en la muestra de la derecha, pues la señal de diente de sierra tarda más tiempo en alcanzar el valor de la muestra y por lo tanto en finalizar el pulso. De esta forma se ha conseguido que una señal continua con amplitud variable se convierta en una señal codificada en tiempo y por tanto estas serían el tipo de entradas que se le introducirían teóricamente al filtro propuesto.

Una vez dicho esto, el propósito de nuestro circuito demostrador será entonces reproducir el efecto que tendría en un tiempo discreto el filtrado de una señal codificada en tiempo PWM (Figura 7 parte inferior), el cual, para tener como referencia una aplicación más común y fácil de visualizar, sería equivalente al efecto de un filtrado de una señal codificada en amplitud en un tiempo discreto por un filtro adaptado a ella (Figura 7 parte superior).

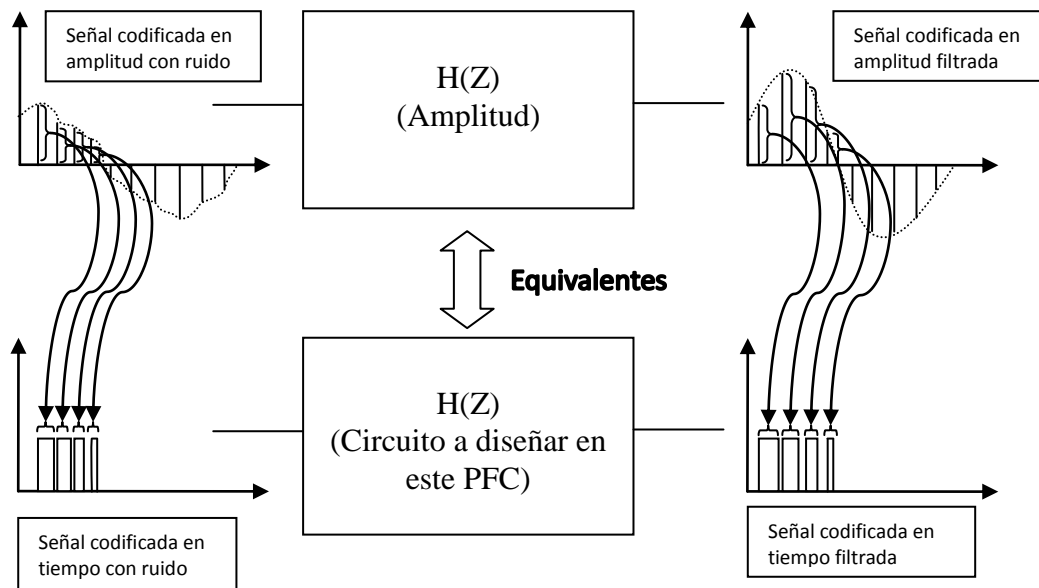


Figura 7: Funcionamiento de los filtros con señales moduladas.

Sin embargo, a la hora de comprobar el efecto del filtro en la señal, para tanto simular como probar en el laboratorio nuestro circuito, en lugar de introducir señales reales moduladas, introducimos una señal de tren de pulsos de ancho constante. El motivo de esto es debido a que si se le introducen este tipo de señales al filtro lo que se hace es obligar al mismo a actuar como integrador, en este punto de funcionamiento el filtro debería funcionar obteniendo a su salida un tren de pulsos que en este caso crecerá exponencialmente con el paso del tiempo como respuesta al tren de pulsos constante a su entrada. Esto sería equivalente a lo que ocurriría al introducir una entrada escalón a un filtro continuo de primer orden, donde a su salida se obtendría una señal que crece exponencialmente hasta alcanzar la amplitud máxima de la señal escalón que se le ha introducido. El motivo de la equivalencia es muy simple, si pensamos detenidamente, si se codifica una señal continua (escalón) en PWM lo que obtendríamos sería un tren de pulsos de ancho constante. Esto resulta bastante lógico ya que la amplitud de una señal escalón continua es invariable y si la introducimos al modulador de ondas PWM las muestras tomadas por el circuito muestreador tendrán en todo momento la misma amplitud, la consecuencia de esto será que al compararlas con el diente de sierra a la salida del comparador la señal tendrá siempre el mismo ancho de pulso.

Por lo tanto, esta es una forma de comprobar el funcionamiento del sistema sin tener que añadir un circuito modulador ni señales moduladas a su entrada, ya que sabemos cómo ha de ser la respuesta que tiene que dar el filtro ante una entrada conocida, simplificando así la batería de pruebas a realizar. Una imagen de cómo funcionaría el circuito como integrador se puede ver en la figura 8, a la izquierda se podría ver el funcionamiento de un filtro continuo integrando mientras que a la derecha se vería su equivalente en filtro discreto de señales PWM. En ambas se puede apreciar un crecimiento exponencial en la señal de salida.

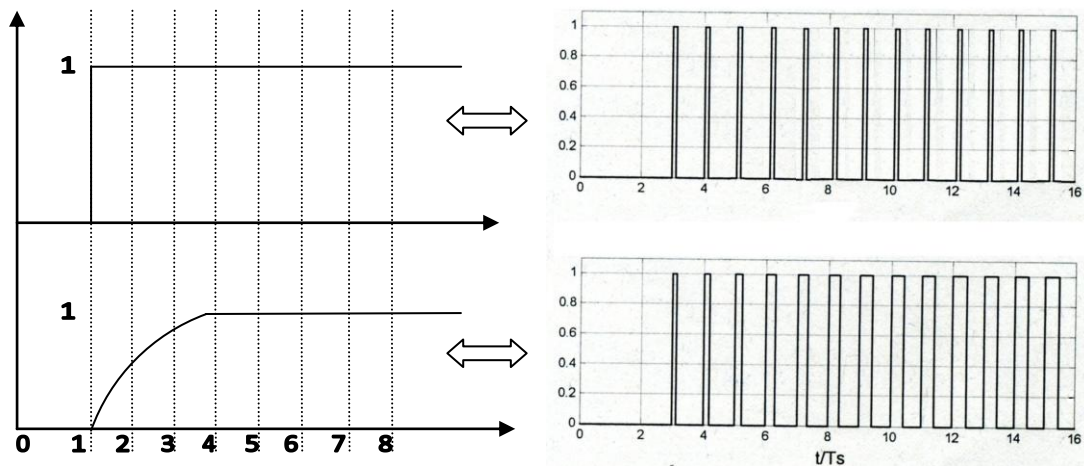


Figura 8: Funcionamiento de ambos filtros funcionando como integrador.

Una vez explicadas las señales de entrada que se le introducirán al filtro y su funcionamiento a grandes rasgos, se puede hablar de cómo estará diseñado a nivel de bloques y circuito. El filtro definitivo construido será de primer orden debido a que proporciona una cierta simplicidad a la hora de entender el funcionamiento del circuito y el principio de funcionamiento del proyecto. Como se puede observar en la figura 9, un filtro de paso bajo discreto de primer orden teóricamente consta de 3 bloques: un sumador, una célula de retraso, y un coeficiente alfa para aplicar una cierta ganancia al sistema. Si se unen como se ha realizado en la figura se forma un filtro paso bajo discreto.

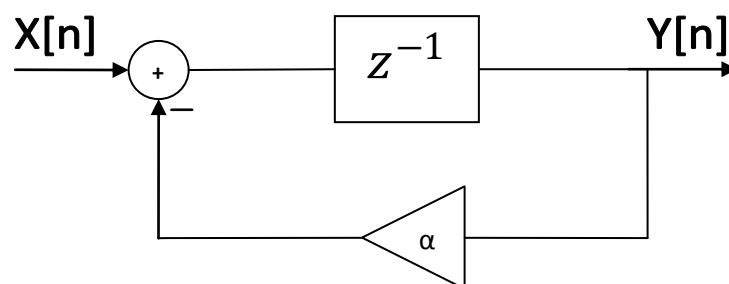


Figura 9: Diagrama de bloques de un filtro de paso bajo discreto.

El funcionamiento de un filtro paso bajo con respecto a la frecuencia de la señal de entrada puede verse en figura 10, a las señales que tienen bajas frecuencias el filtro las deja pasar intentando que su ganancia sea 0dB para no modificarlas, mientras que las señales que tienen frecuencias altas se atenúan intentando que a la salida del filtro su presencia se anule. Por ello a la hora de diseñar un filtro paso bajo, o cualquier otro tipo de filtro, hay que tener en cuenta dos parámetros primordiales de éste, su ganancia y su frecuencia de corte, como hemos dicho la ganancia interesa que sea de 0dB para no modificar la amplitud o el tiempo de la onda que queremos filtrar y que sea lo más fiel posible a la onda original (aunque esto puede no ser así según qué intenciones tengamos para el filtro), en cambio la frecuencia de corte sí que interesará modificarla según las frecuencias que tengan la señal original y su ruido. La forma de modificar la frecuencia de corte en un filtro es modificando la posición de su polo (1 único polo si el filtro es de primer orden), pues a partir de la frecuencia en la que éste se encuentre el filtro

empezará a atenuar las señales de esa frecuencia o superiores, y por lo tanto a partir de esa frecuencia es cuándo empezará a filtrar.

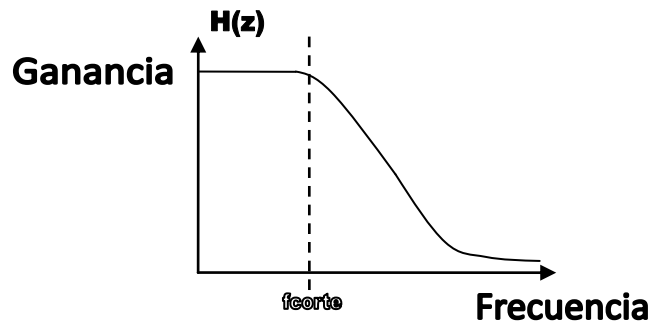
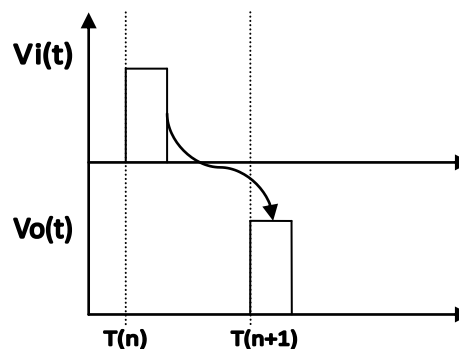


Figura 10: Respuesta en frecuencia de un filtro de paso bajo.

En el siguiente apartado se explicará detalladamente el funcionamiento de la célula de retraso, puesto que es el bloque clave a la hora de crear el circuito.

2.1.- La célula básica del circuito, la célula de retraso.

Como se puede observar en la figura 9, una de las partes que forman el filtro de paso bajo es una célula que retrasa la señal un periodo (z^{-1}), es decir, coge la señal de entrada en un periodo actual y la repite un periodo más tarde.



Esta es la parte del circuito más importante y a la vez más compleja de entender, por ello se explicará detenidamente.

Como se ha explicado, la célula de retraso se encarga de copiar un pulso de entrada justamente un periodo después para así poder realimentarlo como se muestra en la figura 9 y completar el filtro de paso bajo, podríamos decir entonces que la célula actúa como una memoria respecto al circuito, ya que retiene un pulso dado con anterioridad. Para realizar esta copia se ideó un circuito (Figura 11) que usaría una serie de circuitos RC como una parte del sistema global, para explicar el funcionamiento de éstos se considerará que todos ellos comienzan con una carga en los condensadores $V_{C1}=V_{C2}=V_{C3}=0V$ para así poder empezar teniendo una referencia de tensiones. Además, en un régimen transitorio, interesará que todos ellos tengan la misma constante de tiempo RC o también llamada τ (la cual expresa a la velocidad a la que se cargan los condensadores), ya que de esta forma se cargarán todos a una misma velocidad, siendo

esta característica clave para facilitar la explicación de cómo se retrasa el pulso de entrada.

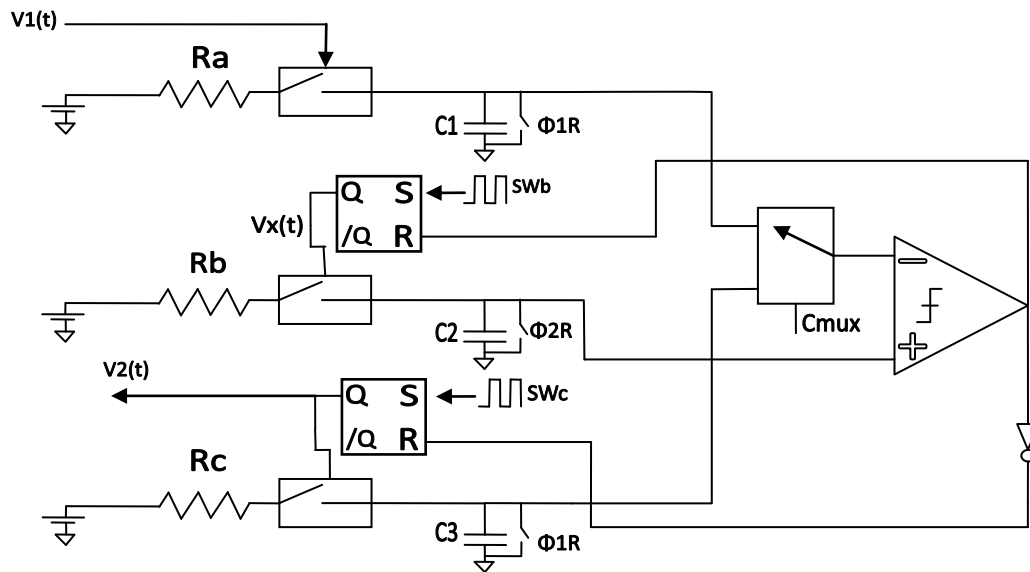


Figura 11: Esquema de bloques de la célula de retraso

Una vez representado el diagrama de bloques, es importante puntualizar que este circuito sólo actúa con valores positivos de amplitud, lo cual en realidad no es ningún problema ya que las señales de entrada al circuito que como hemos dicho serán señales PWM o trenes de pulso de ancho constante, para el caso de nuestro demostrador únicamente variarán su tensión entre 0 y 5 voltios y por lo tanto la célula de retraso sólo necesitará abarcar señales positivas, simplificando cuantiosamente el diseño de la misma.

En el siguiente cronograma se explicará el funcionamiento de la célula de retraso y cómo actúan las ondas tanto externas como internas del circuito. Pero antes de explicar esta secuencia de funcionamiento es importante saber de qué forma se carga un condensador mediante una resistencia, o lo que es lo mismo la expresión de cómo funciona la carga de un circuito RC. La tensión entonces a la que se carga el condensador a través de una resistencia cuando se le introduce una tensión continua a su

entrada vendrá dada por la siguiente ecuación: $V_{C1}(T_s) = V_r \left(1 - e^{-\frac{T}{\tau}} \right)$ [Ec.1]. En esta

expresión se puede observar que el tiempo de carga del condensador (T) para una determinada tensión depende de la propia constante de tiempo RC (τ) que tenga el circuito.

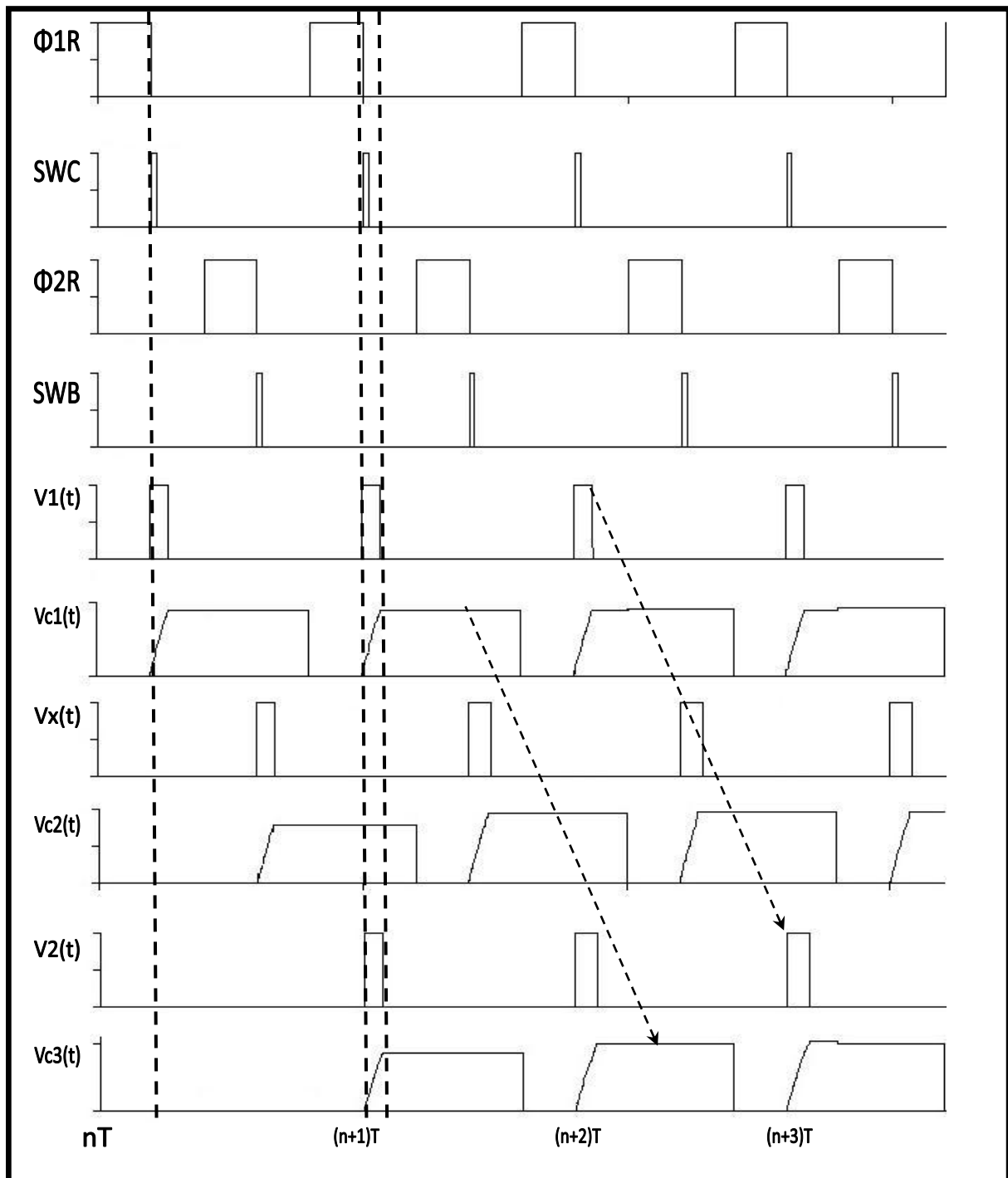


Figura 12: Cronograma de funcionamiento de la célula de retraso.

Ahora ya sí se puede comenzar a explicar el funcionamiento de la célula ya que tenemos todo lo necesario para entenderlo. Para empezar, se podría decir que cada periodo de funcionamiento del circuito consta de 4 fases completamente definidas, el tiempo de duración de la primera fase estaría dedicado al borrado de los condensadores C_1 y C_3 para que comiencen cargados a 0 voltios, puesto que en la siguiente fase, la fase dos, llegará el momento de que éstos tengan que actuar recibiendo distintas señales de carga. La tercera y cuarta fases serían análogas a primera y segunda fases, pero actuando sobre el condensador C_2 , ya que éste funcionará siempre de intermediario entre periodos. Cada una de las frases se explicará detenidamente.

Como se puede ver en el cronograma, la primera señal que actúa sobre el circuito es la denominada $\Phi 1R$, ésta lo que hace es activar un interruptor que conecta directamente los condensadores C_1 y C_3 con tierra haciendo que se elimine cualquier carga que pudieran tener anteriormente y consiguiendo así éstos que empiecen con 0V para tener una referencia como se ha mencionado con anterioridad.

Una vez descargados, la señal de entrada entonces al circuito $V_1(t)$, mientras se encuentra a nivel alto, activa un interruptor que hace que se cargue el condensador C_1 a una determinada tensión según la ecuación descrita anteriormente [Ec.1]. Tras la primera carga realizada por la señal de entrada al circuito, y justo a la mitad del periodo entre T_n y T_{n+1} , el condensador C_2 es borrado como se hizo antes con el condensador C_1 , pero esta vez mediante la señal $\Phi 2R$, tras este borrado una señal de activación (SW_b) conectada a la entrada Set de un biestable RS da un pulso, esto hace que la salida del biestable pase a nivel alto y cierre así el interruptor correspondiente para la carga del condensador C_2 , de esta forma C_2 comienza a cargarse a la misma velocidad que lo hacía el condensador C_1 al tener la misma constante de tiempo. La carga de este condensador C_2 continuará hasta que la tensión de ambos condensadores sea la misma, momento en el cual el comparador actuará y cambiará su salida pasando de estar en nivel bajo para estar a nivel alto, esta salida está conectada a la entrada Reset del biestable RS y por lo tanto al haber un pulso positivo en el mismo, la salida del biestable permutaría a 0 voltios, lo que desembocaría entonces en la apertura del interruptor que conecta la tensión de alimentación de los condensadores con el circuito RC y con lo cual se mantendría una carga constante en el condensador consiguiendo así que C_1 y C_2 tengan la misma carga, si durante todo este proceso se observa la salida del biestable RS vista en el tiempo, es decir su salida Q ($V_x(t)$), se puede observar que ésta tendrá el mismo ancho de pulso que la señal de entrada al circuito $V_1(t)$.

De la misma forma actuará el circuito al llegar al periodo T_{n+1} , en este caso, mientras el condensador C_1 recibe una nueva señal, el condensador C_3 se encarga de copiar la carga del condensador C_2 y por lo tanto el mismo ancho de pulso que $V_1(t)$ tenía en el periodo anterior, para ello una señal de activación SW_c se encargará de cerrar el interruptor del circuito de carga para este condensador que igual que en el caso anterior se hará activando un biestable, una vez que la tensión C_3 llegue a la misma que C_2 (que aún mantiene la tensión anterior pues hasta la mitad del periodo ésta no se borrará), como antes, el comparador actuará para activar el Reset del biestable, la señal que obtendremos a la salida de éste será $V_2(t)$, la cual será del mismo ancho de pulso que la señal $V_1(t)$ en T_n , pero en este caso estaremos en T_{n+1} , habiendo retrasado y memorizado la señal un periodo más tarde.

Por otro lado, si observamos el comparador en el diagrama de bloques, se puede apreciar como antes de la entrada al mismo hay un multiplexor que elige qué tensión se debe comparar en el circuito según en la fase en la que estemos. Esto hace que en el caso de la comparación que se realiza entre C_3 y C_2 (si observamos a que entrada van conectadas las tensiones del condensador), se puede ver que en el momento en que C_3 llegue a un mismo nivel de tensión que C_2 el comparador pasará de estar a nivel alto a nivel bajo, pues la señal V_{C2} está conectada en la entrada positiva del comparador y la entrada V_{C3} está conectada a la entrada negativa del mismo. Por lo tanto, para que el comparador sea capaz de mandar el pulso positivo para activar el Reset del biestable, se necesita el uso de un inversor, de ahí que a la salida del comparador haya una conexión

directamente a uno de los biestables, y otra vaya conectada a este inversor, de tal forma que aunque la transición sea de nivel alto a nivel bajo, el inversor la permutará y entonces el circuito será capaz de abrir el interruptor que carga al condensador C_3 . Justo lo contrario pasaría al principio, al tener V_{C2} que copiar a V_{C1} la permutación del comparador será al revés, de nivel bajo a nivel alto, de ahí que no se necesite el inversor para que funcione correctamente el biestable.

Una vez explicado el funcionamiento del circuito que funciona como célula de retraso, sería conveniente describir las ecuaciones que expresan cómo se cargan las tensiones en cada uno de los distintos condensadores del circuito. Para que así se pueda apreciar y calcular la respuesta del circuito ante una o varias señales externas y como afectan a la elección de los componentes.

Así pues las constantes de tiempo τ de cada uno de los circuitos RC serán:

$\tau_{ai} = R_{ai}C_i$ [Ec.2], siendo $i=1 \dots N$, Constante de tiempo para el circuito RC de entrada. Puede haber varios coeficientes de τ_a ya que el condensador C_1 podría ser cargado mediante varias señales de entrada si deseáramos realizar un sistema multibit.

$\tau_c = R_cC_3$ [Ec.3], constante de tiempo del circuito del condensador C_3 .

Por lo tanto podemos definir las tensiones en cada uno de los tres condensadores, que se verán impuestas por las siguientes ecuaciones:

$$V_{C1}((n+1)T_s) = V_r \left(1 - e^{\frac{-T_s}{\tau_a}} \right) \quad [\text{Ec.4}]$$

$$V_{C2}((n+1)T_s) = V_{C1}(n+1)T_s \quad [\text{Ec.5}]$$

$$V_{C3}((n+1)T_s) = V_r \left(1 - e^{\frac{-T_s}{\tau_c}} \right) = V_{C2}(nT_s) \quad [\text{Ec.6}]$$

Analizando estas ecuaciones lo que podemos deducir es que durante un mismo periodo $n+1$, tanto V_{C1} como V_{C2} llegan a una misma tensión. Además, también se puede observar como el valor de la tensión en el condensador C_3 en un periodo $n+1$ es exactamente la misma que la tensión que tenía el condensador C_2 en el periodo n . O lo que es lo mismo, C_3 ha copiado la carga de C_2 con medio periodo de retraso que a su vez era la carga que originalmente tenía C_1 al comienzo del periodo anterior.

¿Qué ocurre cuando a la entrada de la célula de retraso hay varias señales conectadas con una resistencia al condensador C_1 teniendo un sistema multibit?

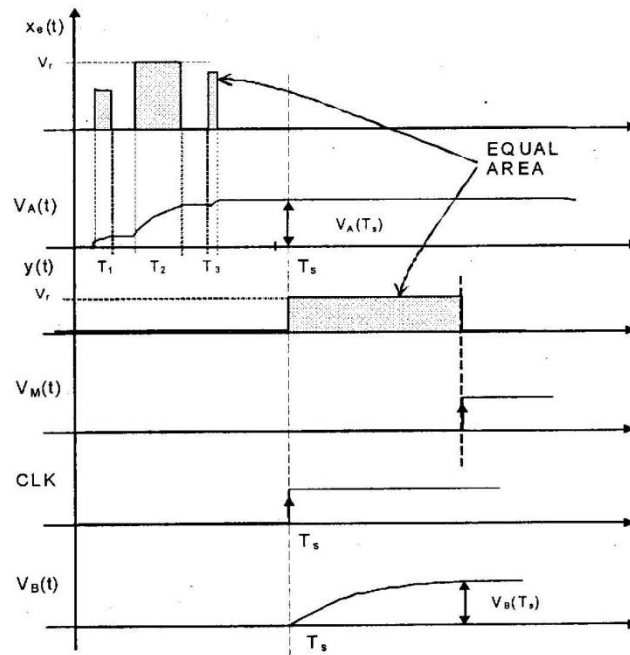


Figura 13: Funcionamiento del circuito si la entrada es multibit (varias entradas).

Si quisiéramos hacer un sistema multibit, a la entrada del circuito podrían haber varios pulsos para un mismo periodo cargando el condensador C_1 a una determinada tensión V_{C1} (en el caso de la imagen $V_A(t)$). Pero como se puede observar en el diagrama de bloques propuesto para la célula de retraso (Figura 11), éste no puede repetir una serie de pulsos separados en el condensador C_2 debido a su diseño, esto en realidad no es un gran inconveniente ya que la señal que se copiará en este condensador tendrá exactamente la misma área que el total de todos los pulsos introducidos en el condensador C_1 , lo cual es matemáticamente muy interesante.

De tal forma podríamos decir que la ecuación que representa al ancho de pulso generado por la señal V_2 ($y(t)$), y que será copia de señal de entrada al circuito y que carga a C_1 , o en el caso en el que se hayan introducido varios pulsos a la entrada del circuito, la señal proporcional a la suma de las n entradas que actúen sobre él, estará representada como:

$$T_y[n] = \sum_{i=1}^N \frac{\tau_c}{\tau_{Ai}} T_{xi}[n-1] \quad [\text{Ec.7}]$$

Siendo T_Y el ancho de pulso de la señal a la salida de la célula de retraso en $[n]$ y T_X el ancho de pulso de la señal de entrada al circuito en un periodo $[n-1]$. El coeficiente $\frac{\tau_c}{\tau_{Ai}}$ [Ec.8] es una constante definida por el valor de los componentes resistivos y

capacitivos de los circuitos de carga que funciona como ganancia de la célula y modifica el ancho del pulso de salida, si su valor es distinto de 1, es decir, si los

circuitos RC no se cargan a la misma velocidad, el ancho de pulso necesario para que el condensador se cargue a una misma tensión en estos circuitos será diferente, si por ejemplo C_3 se carga más rápidamente, el ancho de pulso de salida será menor puesto que necesitará menos tiempo para llegar a la tensión a la que estaba el condensador C_1 , en cambio si la carga es más lenta, el ancho de pulso en la salida será mayor, ya que lógicamente el interruptor necesitará estar activo más tiempo para que el condensador copie la tensión del condensador C_1 .

2.2.- Ejemplo de filtro de 1^{er} orden e integrador.

Una vez explicado el funcionamiento y haber visto el diagrama de bloques del circuito de la célula de retraso, se explicarán los detalles del circuito una vez actúe como filtro e integrador.

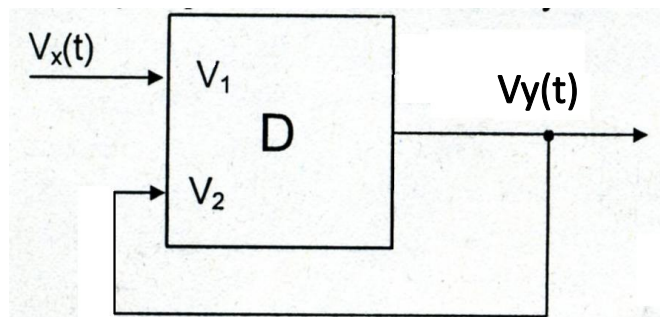


Figura 14: Esquema de filtro de 1^{er} orden simplificado.

La célula de retraso antes explicada puede ser usada para implementar cualquier tipo de filtro de tiempo discreto. Un posible ejemplo para ello es usar la célula de retraso para la formación de un filtro de 1^{er} orden que pueda ser usado como un bloque para construir cualquier filtro lineal arbitrario o un integrador, que es precisamente el objetivo del demostrador construido en el proyecto. La figura 14 muestra como sería el diagrama de bloques a muy alto nivel de un filtro paso bajo unipolar de 1^{er} orden. En esta figura se puede observar que el bloque D, que sería el correspondiente a la célula de retraso (Delay), tiene dos entradas, V_1 será la señal de entrada externa al circuito, mientras que V_2 es la realimentación que viene de la señal retrasada y que cierra el lazo del filtro sobre la unidad de retraso. Esta realimentación al ser una entrada más al circuito necesitará por lo tanto la implantación de un nuevo interruptor y de una nueva resistencia de acuerdo con el diagrama de bloques, de esta forma hay que tener en cuenta para todos los cálculos que habrá un nuevo coeficiente de constante de tiempo.

Por otro lado, si se analiza la función de transferencia del sistema se obtendría la siguiente expresión:

$$H(z) = \frac{V_y(z)}{V_x(z)} = \frac{g \cdot z^{-1}}{1 - p \cdot z^{-1}} \quad [\text{Ec.9}]$$

Tanto la posición del polo del filtro como el valor de la ganancia de la célula de retraso vienen controlados por los coeficientes p y g respectivamente, los cuales se pueden observar en la función de transferencia y vienen dados por los valores de los diferentes condensadores y resistencias del circuito, como se podrá observar en las ecuaciones dadas a continuación.

Usando los coeficientes de los circuitos RC descritos anteriormente $\tau_{a1} = R_{1a}C_1$, $\tau_{a2} = R_{2a}C_2$ y $\tau_c = R_cC_3$, se puede definir la ecuación que expresa ancho de pulso de la señal de salida del filtro T_y , que será:

$$T_y[n] = \frac{\tau_c}{\tau_{A1}} T_x[n-1] + \frac{\tau_c}{\tau_{A2}} T_y[n-1]. \quad [\text{Ec.10}]$$

Como se puede ver, de esta última ecuación se obtienen dos coeficientes: El primero sería $\frac{\tau_c}{\tau_{A1}}$ [Ec.11], el cual acompaña a la señal de entrada $V_1(t)$ retrasada un periodo, siendo T_x el ancho del pulso de la misma y dando así una ganancia a la célula de retraso, así pues este coeficiente será el denominador g en la función de transferencia.

Luego estaría el coeficiente $\frac{\tau_c}{\tau_{A2}}$ [Ec.12], acompañando a la señal de salida $V_y(t)$ y por lo tanto configurando el coeficiente p . La variable τ_{A2} que interviene en p será precisamente debida a la nueva constante RC introducida al circuito por los componentes implantados por la realimentación.

Como se puede observar en las anteriores ecuaciones, ni C_2 ni R_b están involucrados en los coeficientes ya que su única misión es hacer de intermediario entre C_1 y C_3 , lo único que tiene que asegurar con el valor de sus componentes es que va a copiar la tensión de C_1 en lo que dura su fase, por lo tanto su valor estará dictado solo para temas de implementación en el circuito sin afectar a las matemáticas del mismo. Aunque en nuestro caso tendrá los mismos valores que los demás componentes de la célula de retraso para simplificar su implementación.

Volviendo de nuevo al coeficiente p , como se ha dicho, podríamos definirlo como el que representa la posición del polo dentro del círculo unidad y según dónde éste se encuentre en el mismo se definirá la velocidad del filtro que construyamos.

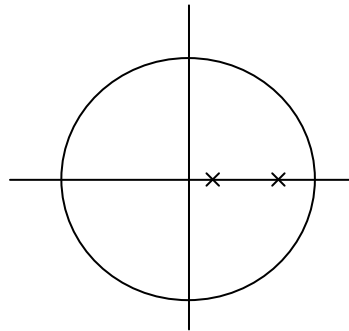


Figura 15: Situación de los polos en el círculo unidad.

En cuanto a esta posición del polo, hay que tener cuidado de que el valor de p no supere a 1, pues como se puede observar, el área de funcionamiento estable del filtro se encuentra en una circunferencia de radio unidad, y si el valor de p supera estos límites, el circuito entrará en un punto de funcionamiento inestable

Así pues y dentro de estos límites, cuánto más cerca esté el polo del eje más rápido será la actuación del circuito, mientras que cuánto más alejado esté de éste, la integración será más progresiva. Esto tiene muchísima importancia ya que influirá críticamente en la elección de los componentes del circuito y en el diseño del mismo. Lo que nos interesa a nosotros es que el filtrado no sea demasiado rápido, ya que de esta forma se podrá ver mejor el funcionamiento del circuito y la modificación e integración que realiza a la onda. En la figura 16 se puede ver gráficamente la reacción de un sistema de primer orden a la posición del polo en sus ejes.

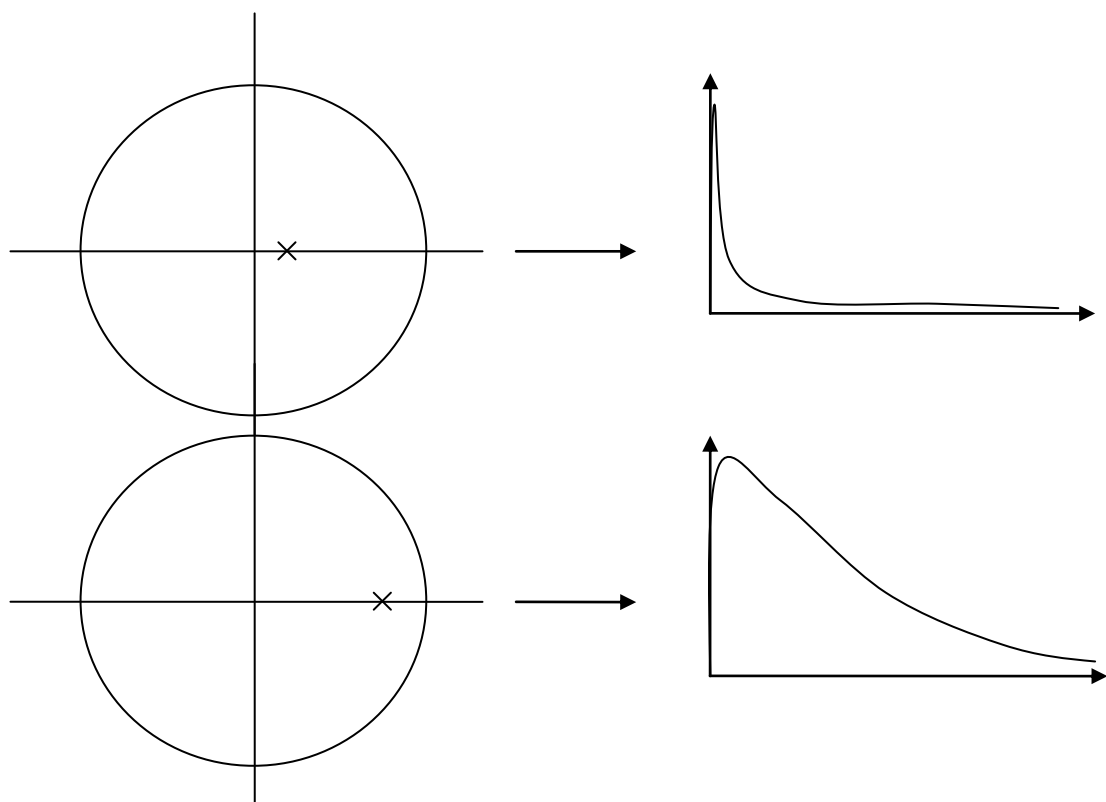


Figura 16: Respuesta del sistema según la posición del polo.

Para observar cómo sería el funcionamiento del integrador y ver una aproximación de lo que haría, en la siguiente gráfica se muestra lo que ocurre si por ejemplo se introduce una señal escalón en $n=2$, usando coeficientes $g=1$ y $p=0,8$. Se puede ver como el filtro funciona como integrador pues la señal crece exponencialmente hasta que alcanza su ancho de pulso máximo.

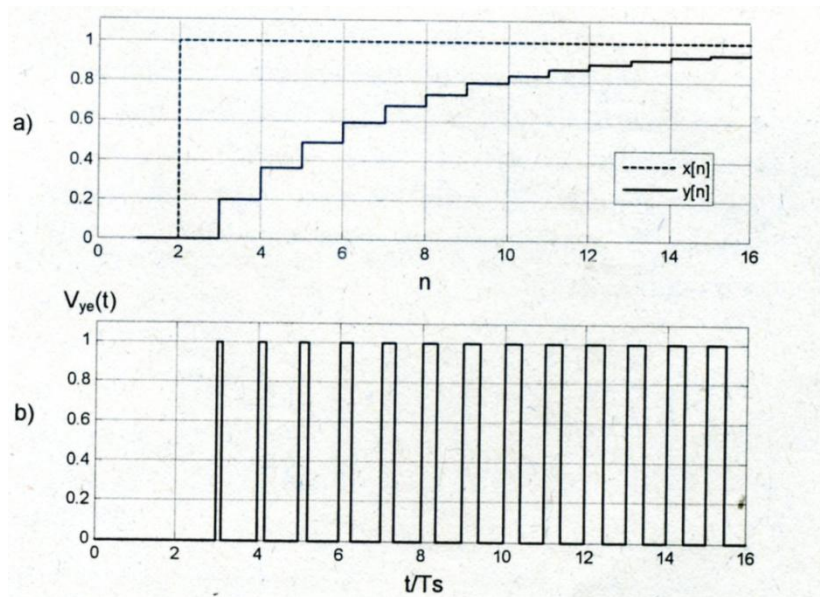


Figura 17: Funcionamiento del circuito como integrador para una $g=1$ y un $p=0.8$.

De esta forma, en este subapartado se ha explicado el funcionamiento del filtro y sobre todo la importancia que tiene el valor de los componentes en el mismo.

2.3.- Diagramas de bloques del demostrador.

En los diagramas de bloques se mostrarán dos diferentes casos que se han utilizado en el proyecto a la hora de simular y diseñar el filtro, el primero es un diagrama de bloques a bajo nivel usado para simular el circuito mediante el programa Cadence OrCAD 15.7, mientras que el segundo corresponde a un diagrama de bloques que esquematiza al circuito finalmente construido.

Ahora es el momento de explicar a nivel de diagrama de bloques y circuito como sería el filtro definitivo y su funcionamiento como se hizo con el bloque de retraso. En realidad, el funcionamiento del circuito como filtro no dista mucho de su funcionamiento como simple célula de retraso, el principio de trabajo de copia de pulsos y de carga de los condensadores mediante circuitos RC sigue siendo exactamente igual a como era cuando funcionaba como célula, así como las fases y las señales de control que la manejaban, pero sin embargo se añade un sistema de realimentación al circuito que hace que los pulsos ya no sean copias exactas de la señal de entrada al mismo, si no que la salida será una señal PWM perfectamente integrada y filtrada.

2.3.1.- Diagrama de bloques del demostrador simulado.

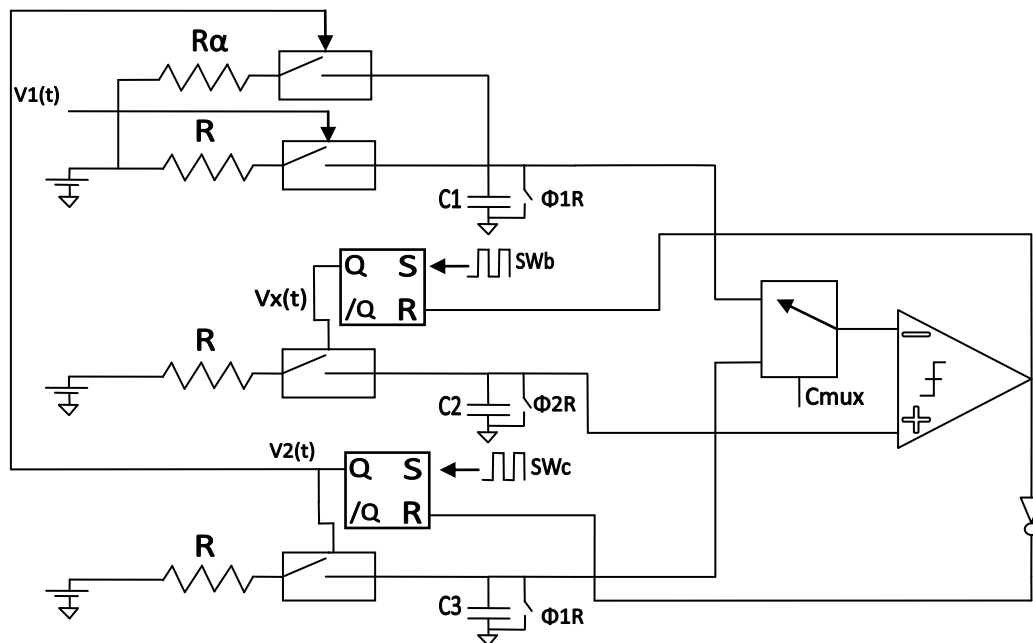


Figura 18: Esquema de bloques para simulación.

Como se ha explicado en el subapartado anterior, para comprobar el funcionamiento del circuito, se propuso que el demostrador se configurara como filtro de paso bajo de 1^{er} orden, haciendo que hubiera una realimentación a su salida sin ningún tipo de entrada múltiple, con lo cual el condensador C_1 sólo sería cargado por dos posibles señales que cerraran el circuito RC. A diferencia de la célula de retraso, en este diagrama de bloques existe una realimentación a un nuevo interruptor que se ha añadido y que cierra el circuito mediante una nueva resistencia, esto en realidad es el bloque sumador representado en la figura 9, ya que en el momento en el que haya una entrada al circuito ésta se sumará con el pulso de realimentación mediante el paralelo de las dos resistencias, este paralelo hará el circuito más rápido (ya que la resistencia total es menor y entonces la constante RC será más rápida) y por lo tanto hará que el condensador se cargue a una mayor tensión durante el mismo periodo de tiempo que si sólo le entrara una única señal. Esto causará que el circuito RC del tercer condensador, al tener una τ_c constante, necesite un mayor tiempo para alcanzar a la tensión a la que se ha cargado el condensador C_1 que como hemos dicho, con la realimentación se ha cargado a una mayor tensión al tener una τ menor. Este mayor tiempo necesario para copiar la tensión se traducirá en un pulso más ancho a la salida del biestable y por lo tanto en una señal a la salida del filtro que crecerá exponencialmente con el paso de los periodos.

En este diagrama las señales de control vistas en el cronograma (Figura 12) tales como SW_b o $\Phi 1R$ vienen definidas por fuentes de tensión de onda cuadrada proporcionadas por el simulador y el “capturador” de esquemáticos, las cuales están configuradas para que tengan un ancho de pulso determinado y se pongan a nivel alto en un momento oportuno del periodo tal y como están representadas en el cronograma.

Como se puede ver en la figura 18, los interruptores de descarga del condensador están situados de forma que no pasan por ninguna resistencia a la hora de descargarse sino que van conectados directamente a tierra, de tal forma que a la hora de descargar los condensadores, esta descarga se haga de forma prácticamente instantánea en lugar de mediante una resistencia, que lo que haría es que la descarga se hiciera de forma exponencial, esto es importante ya que si la descarga no es lo suficientemente rápida, podría haber problemas de offset en el circuito y errores en la memorización de señales, ya que si el condensador conserva tensión residual por una mala descarga a tierra, el circuito lo interpretaría como si hubiera entrado un pequeño pulso al mismo y por lo tanto las señales copiadas no serían correctas.

Otro concepto importante que hay que resaltar es que las señales $V_{C1}(t)$ y $V_{C3}(t)$ del circuito van conectadas a un multiplexor, cuya existencia ya se ha mencionado en la explicación de la célula de retraso pero no se ha explicado muy bien el porqué de su implementación. Lo que se consigue con él es ahorrar en el circuito la presencia de otro comparador, con el multiplexor se puede hacer que el comparador actúe según sea requerido comparar unas señales con otras en los momentos del periodo que sean necesarios, por ejemplo, primero para que $V_{C2}(t)$ copie a $V_{C1}(t)$ se necesita que al comparador llegue la tensión $V_{C1}(t)$, pero en otro momento del periodo se necesitará que se compare $V_{C3}(t)$ con $V_{C2}(t)$ y por lo tanto será necesario que al comparador le llegue la tensión $V_{C3}(t)$, gracias al multiplexor un único comparador puede hacer eso sin la necesidad de instalar otro aparte. La única señal que se mantiene siempre a la entrada del comparador es $V_{C2}(t)$ debido a que es la única que se compara con las otras dos tensiones y por lo tanto siempre deberá estar conectada a este componente.

Por último explicar que las salidas del comparador, tanto la normal como la negada al igual que en la célula de retraso, están conectadas al reset de los biestables, de esta forma cuando la salida del comparador los active, la salida del biestable permutará a un nivel bajo y por lo tanto se abrirán los interruptores de carga, haciendo así que los condensadores mantengan una tensión constante.

2.3.2.- Diagrama de bloques del circuito construido.

En este sub apartado se mostrará el diagrama de bloques simplificado del circuito finalmente construido, lo realmente importante en este circuito es la inclusión de un microprocesador para la creación de las señales de control del sistema, ya que coordinar absolutamente todas las señales de control mediante generadores de funciones y ondas TTL sería una tarea extremadamente difícil, además, el microprocesador da la ventaja de que se puede controlar todo mediante software, con lo cual nos da un mayor control sobre el funcionamiento del circuito. Otra ventaja de este circuito es que se puede ahorrar la existencia de los biestables, simulando su funcionamiento por software gracias al microprocesador.

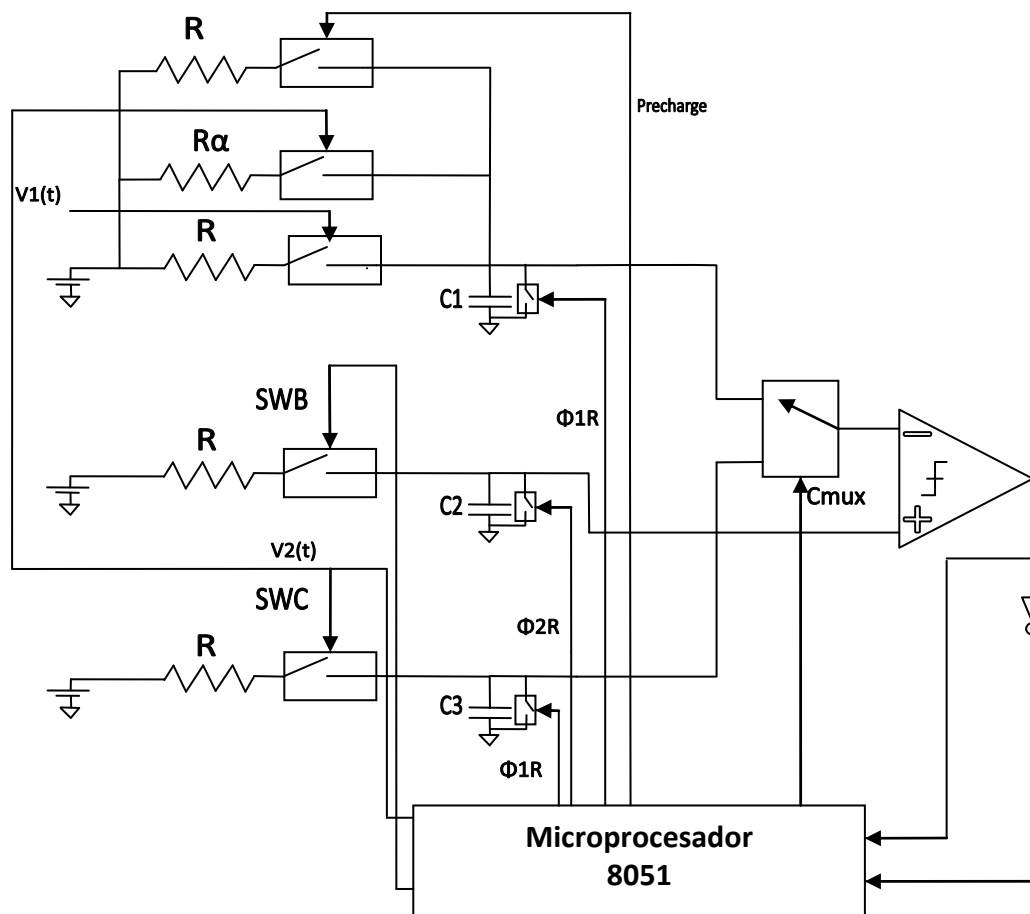


Figura 19: Esquema de bloques del circuito construido.

Las señales de salida del micro se crearán mediante sus puertos E/S y se encargarán tanto de cargar como descargar los condensadores según convenga en el circuito, mientras que las señales de entrada como la salida del comparador irán conectadas a los puertos de interrupciones externas.

A primera vista se puede apreciar como hay varios cambios en este diagrama de bloques con respecto al anterior, a parte de la inclusión del microprocesador, se ha incluido un nuevo interruptor al condensador C_1 y además se han eliminado los biestables RS.

La inclusión de un nuevo interruptor se debe a que así se puede usar uno de los puertos de salida del microprocesador para simular una entrada de tren de pulsos al circuito creada por software (señal precharge), de esta forma es más fácil poner al circuito a funcionar exactamente en el punto de trabajo que se desee, si en lugar de simular una entrada por software se introduce una señal externa (también se puede hacer si se desea), hay que tener en cuenta que hay que sincronizarla con las señales de reloj del circuito, dando cierta dificultad a la hora de comprobar su funcionamiento.

La eliminación de los biestables se debe simplemente a que su trabajo lo realiza el microprocesador. A éstos se les activaba mediante la señales SW, un pulso de estas señales en la entrada Set del biestable hacía que pusieran su salida a nivel alto y un

pulso realizado por el comparador en la entrada Reset hacía que el biestable permutara su salida a nivel bajo. Lo que se hace con el microprocesador es simplemente activar la salida de uno de sus puertos cuando lo haría la correspondiente señal SW, tras esto esperar a que se realice la comparación para desactivarla, esta comparación al realizarse hace que el comparador permute su tensión de salida (esto es a lo que se le llama dar un flanco). Las salidas del comparador pues, tanto la negada como la no negada van conectadas a las entradas de interrupción externa del microprocesador, las cuales son precisamente activadas por flanco y por lo tanto éste es capaz de reconocer los cambios de nivel del comparador como que el condensador ha alcanzado la tensión deseada poniendo así la señal SW a nivel bajo, eliminando así la necesidad de instalar biestables ya que la señal SW será directamente la copia del pulso.

Estos cambios son las únicas variaciones de funcionamiento entre el circuito simulado y el circuito final construido, los demás principios de funcionamiento se mantienen igual. En la siguiente figura 20 se verá cómo funcionan las señales en este circuito mediante un cronograma.

En este cronograma del funcionamiento de la célula de retraso en el circuito construido, es visible lo que hemos explicado, los pulsos de activación de los biestables que había en el anterior y como se ha dicho han desaparecido, ya que toda esa lógica es sustituida por el microprocesador y éste no necesita ningún pulso de activación para funcionar sino sólo programación mediante software, así pues la señal de salida del circuito será por lo tanto SW_c en lugar de la salida de un biestable, que es la misma que abre y cierra el circuito interruptor, pues cuando el microprocesador pone su puerto SW_c a nivel alto se cierra el interruptor correspondiente para que el condensador empiece a cargarse, entonces espera a que haya un flanco dado por el comparador en su interrupción externa para entender que la carga ha sido realizada y entonces permuta a nivel bajo la salida de su puerto, habiendo copiado pues la señal de entrada al circuito y dejando incluso un circuito más fácil de entender que anteriormente, ya que el propio microprocesador es el que se encarga de cerrar y abrir el interruptor con una única señal de uno de sus puertos de salida, sin necesidad de tener un biestable que lo haga y dos señales extra que los activen.

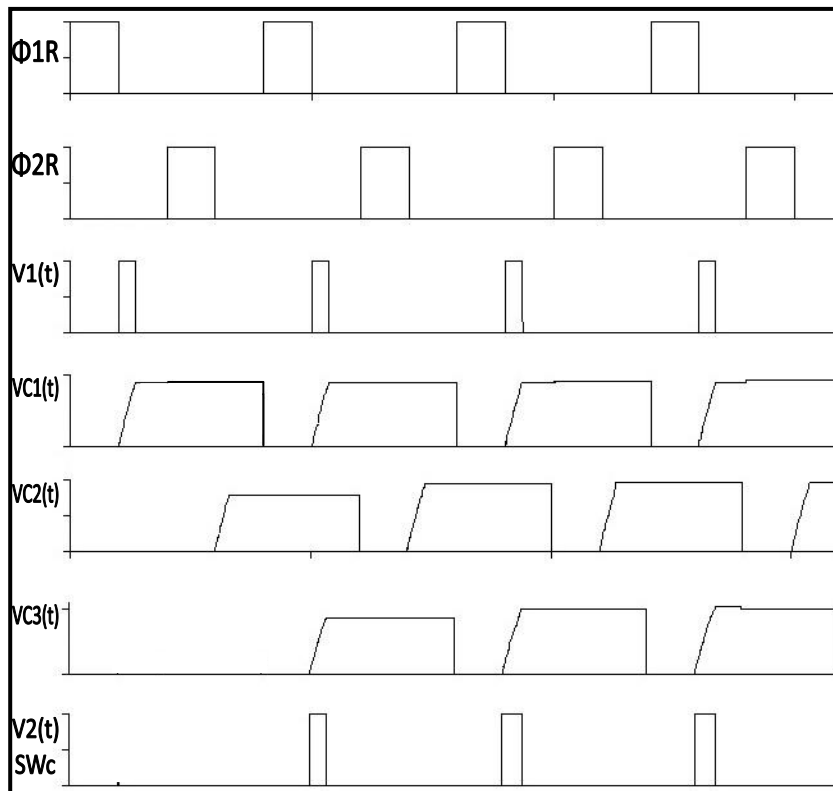


Figura 20: Cronograma de funcionamiento de la célula de retraso en el circuito construido.

2.4.- Elección de componentes semiconductores y microprocesador.

En esta sección se introducirán los componentes elegidos para su implementación en el circuito.

A la hora de elegir los componentes para el circuito, al ser un demostrador, en lugar de centrarnos en la velocidad de funcionamiento del mismo lo hicimos principalmente en la facilidad a la hora de encontrar los componentes discretos además de que éstos se parecieran funcionalmente en tecnología a los que se implementarían en un futuro chip, a partir de estos criterios, elegimos los siguientes componentes:

Interruptores: Para la elección de los circuitos de interruptores se optó por los interruptores CD4066B, principalmente porque son capaces tanto de manejar señales analógicas como digitales, además muy útiles para implementar tecnología CMOS, ya que el chip final que se hará tras este demostrador está basado en esa tecnología.

Otra característica importante para el uso de este integrado es la capacidad para 4 interruptores que tiene cada uno de ellos, ya que para el condensador C_1 , por ejemplo serían necesarios estos 4 como se puede observar en el diagrama de bloques del circuito construido, debido a la gran cantidad de señales que actúan sobre él. Además, tanto su baja resistencia cuando está activado de cómo máximo 250 ohmios como su alta velocidad de actuación gracias a la tecnología CMOS hicieron que la balanza se decantara hacia este tipo de interruptores.

Multiplexor: El multiplexor elegido fue el CD4053, así como con los interruptores, la tecnología CMOS jugó un papel muy importante, ya que al ser este circuito un demostrador de cómo será el circuito implantado en chip con tecnología CMOS, da una visión más realista sobre el funcionamiento del futuro circuito. También hay que tener en cuenta la facilidad para manejar este multiplexor con un microprocesador con las señales de control.

Comparador: Para que el circuito funcione correctamente, es de gran importancia que el comparador no genere errores de Offset en la señal, esto es, que debido a una comparación lenta, se cree una señal un poco más ancha de lo que debería. Para evitar este error se usa el comparador LM311, al ser un comparador de muy alta velocidad, actúa en cuanto la señal se alcanza la tensión de la otra.

Para ilustrar como sería el error de Offset si se usara un comparador lento se incluyó una serie de simulaciones y gráficas que lo muestran para hacerlo más fácil de comprender.

Comparación sin LM311 en la célula de retraso.

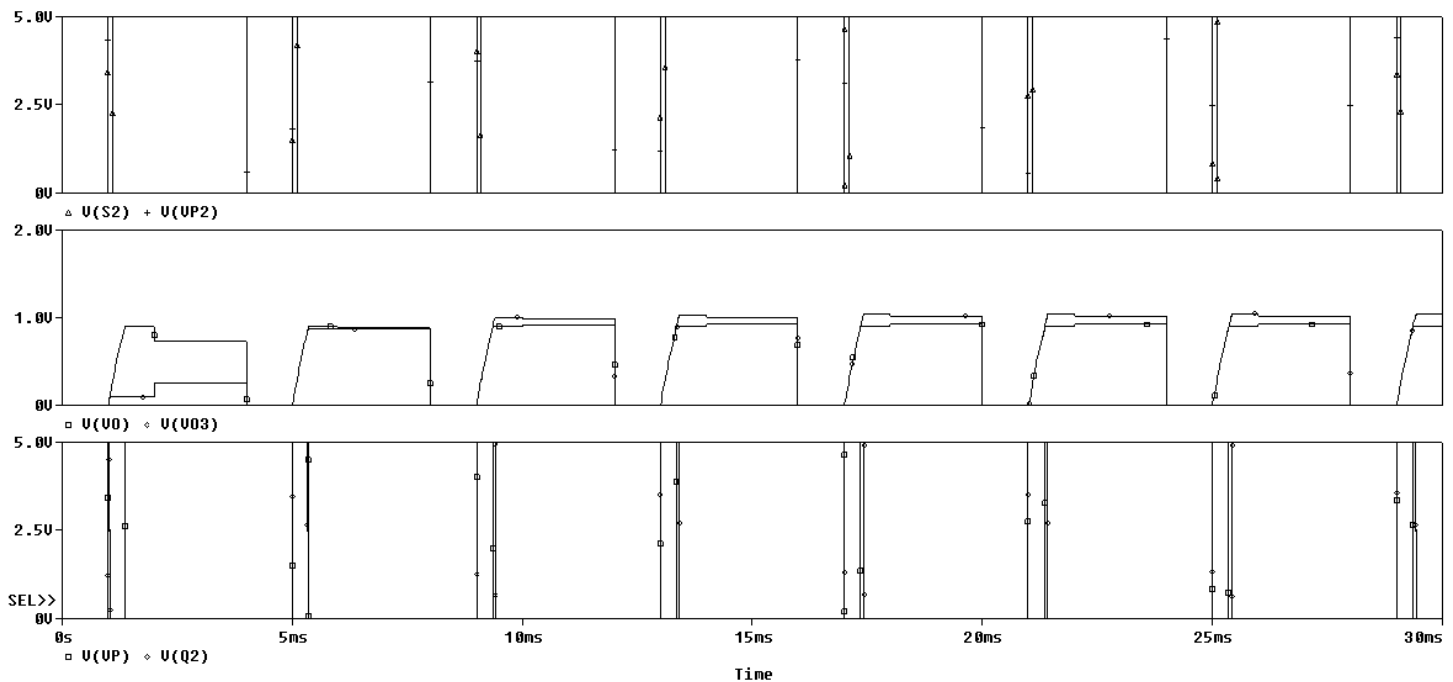


Figura 21: Simulación de la célula de retraso con un LM324.

Como se puede apreciar, la señal de entrada VP es una simple onda cuadrada de un ancho constante, la señal VQ2 marcada con el símbolo del rombo es la señal retrasada 1 periodo respecto a la VP. Esta señal VQ2 cada vez se va haciendo un poco más ancha sin que ese sea el objetivo de esta célula del circuito. Como se puede apreciar en las cargas de los condensadores en la gráfica central, la comparación no es exacta, La señal VO (marcada con el símbolo del cuadrado) sería la tensión de carga del condensador C_1 mientras que VO3 (marcada con el símbolo del rombo) la tensión de carga del condensador C_3 . La señal VO3 no concuerda exactamente con la carga de VO como debería, se puede apreciar que se carga a una mayor tensión, esto es debido a una comparación lenta, el comparador no actúa instantáneamente cuando ambas señales

quedan a la misma tensión y debido a ello permite que estos condensadores adquieran una pequeña carga extra, obteniendo así un error de Offset. El comparador utilizado para esta simulación es un LM324 corriente en lugar de un verdadero comparador de alta velocidad.

Comparación con LM311 en la célula de retraso.

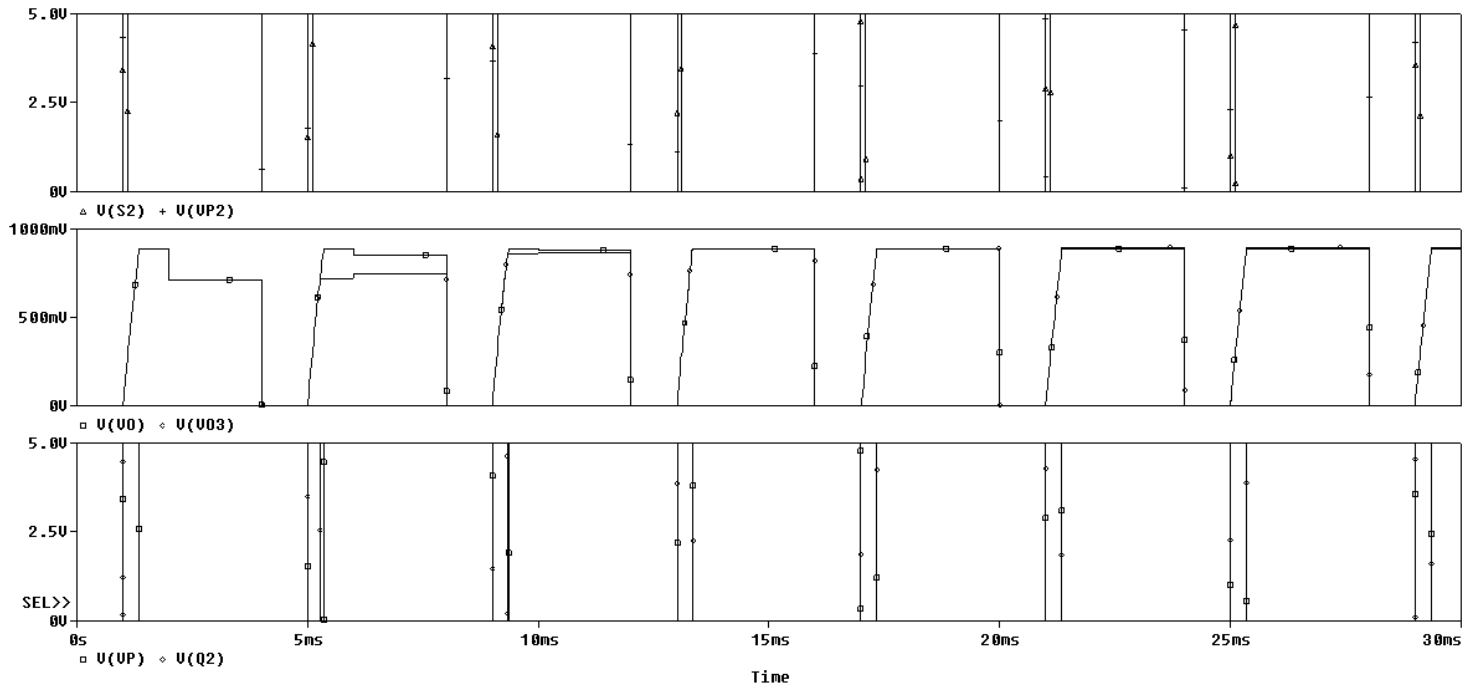


Figura 22: Simulación de la célula de retraso con un LM311.

Como se puede ver con el comparador LM311, el ancho de ambas señales es idéntico tras el primer periodo, eliminando así el factor de error que se introduciría usando un comparador de baja velocidad. Ambas señales cuadradas quedan perfectamente superpuestas al tener el mismo ancho de pulso, a su vez, la carga en los condensadores, como se puede apreciar, son completamente iguales una vez el simulador PSpice se estabiliza, ya que en cuanto las tensiones son iguales, el comparador actúa de una forma casi instantánea.

El microprocesador:

El microprocesador utilizado fue el C8051F330, montado sobre un Toolstick (Figura 23) para su sencillo acceso y programación mediante cualquier PC. La elección del microprocesador de la familia Intel 8051 fue principalmente por su anterior uso durante la carrera en diversas asignaturas, estando así más familiarizado su uso, además de la gran cantidad de puertos de propósito general que tiene para su utilización y dos entradas de interrupciones externas, justo las necesarias para implementar la funcionalidad del circuito final.

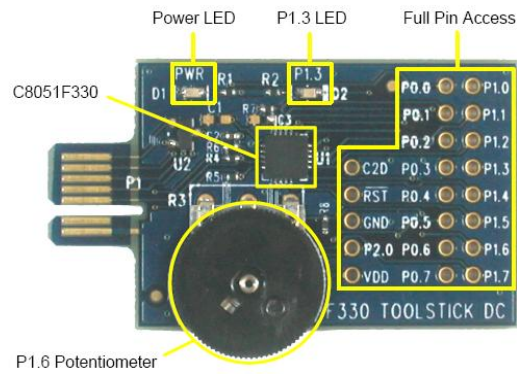


Figura 23: Toolstick en el que va incluido el microprocesador C8051F330

La elección de los componentes resistivos y capacitivos se explicará en el siguiente apartado ya que su implantación en el circuito no es arbitraria sino siguiendo una serie de criterios que se explicarán en el capítulo sobre diseño de hardware.



3.- Diseño de hardware.

En este capítulo de la memoria del proyecto, se explicará principalmente el funcionamiento de los esquemas eléctricos que se usaron en la realización del circuito, tanto el esquema utilizado para realizar las simulaciones del circuito en el simulador como el esquema realizado para crear la placa PCB del mismo, además, en este capítulo se encuentran las simulaciones del circuito, que vendrán explicadas detalladamente para entender su funcionamiento ante distintos tipos de entradas y distintas situaciones.

3.1.- Esquemas eléctricos.

3.1.1.- Esquema eléctrico de simulación del circuito.

Como se ha explicado al comienzo de este capítulo y como se ha podido ver en los diagramas de bloques, hay dos circuitos ideados para el desarrollo y creación del proyecto, uno dedicado a simulación mientras que otro a la construcción del mismo, como antes, primero comenzaremos explicando el circuito eléctrico usado para la simulación, mientras que luego explicaremos el circuito usado para la construcción.

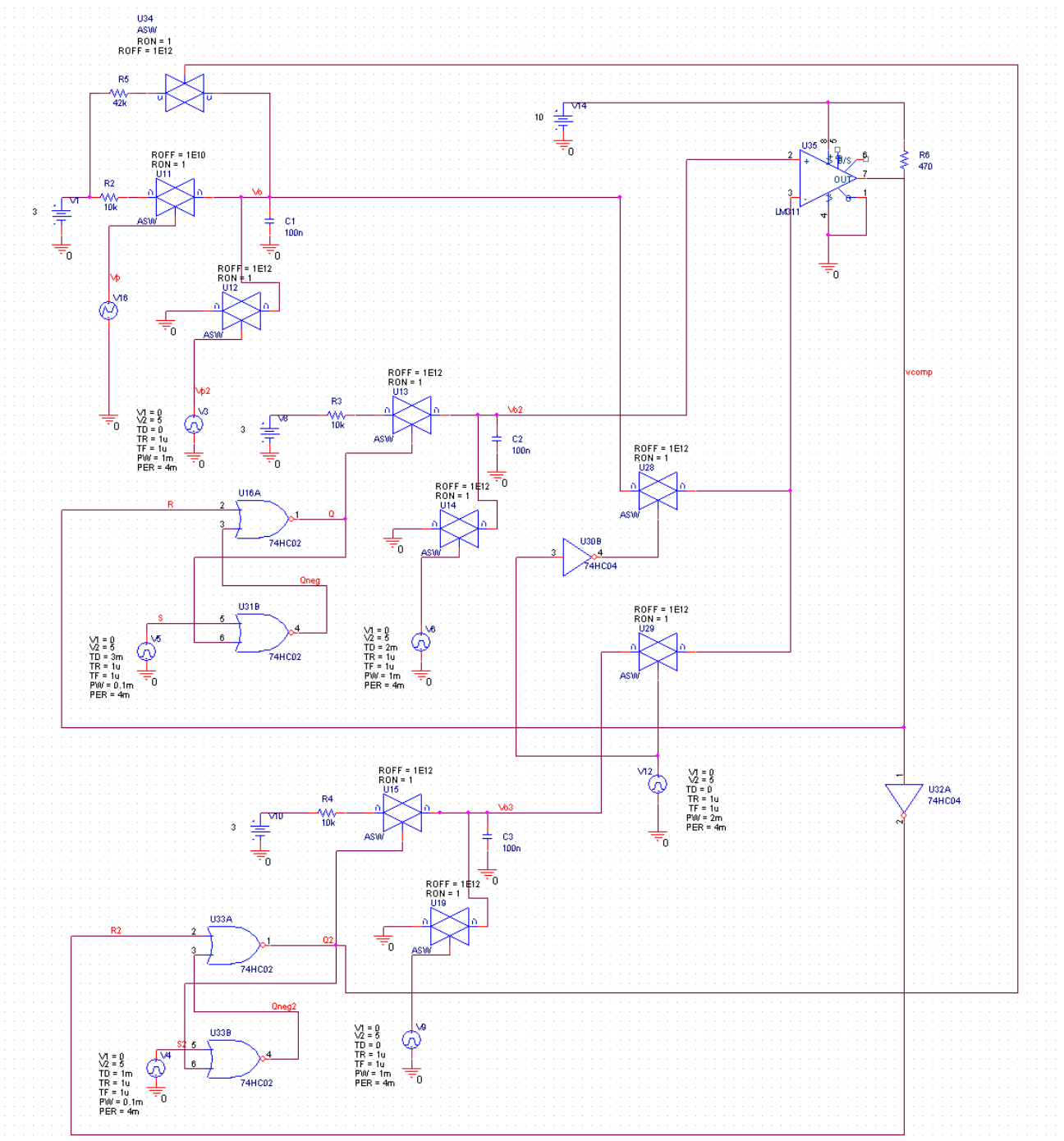


Figura 24: Esquema eléctrico del circuito simulado.

Este primer esquema eléctrico corresponde al diagrama de bloques de simulación del apartado 2.3.1. El motivo por el que no se usa el mismo esquemático tanto para la simulación como para la fabricación del circuito es por motivos técnicos debido al software de diseño, principalmente porque la mayoría de los componentes elegidos para la fabricación del circuito no estaban correspondidos con librerías en el editor de esquemáticos ni en el editor de layout, por lo tanto, para la simulación del circuito se creó un esquema funcional similar al que se ideó en un primer instante, pero

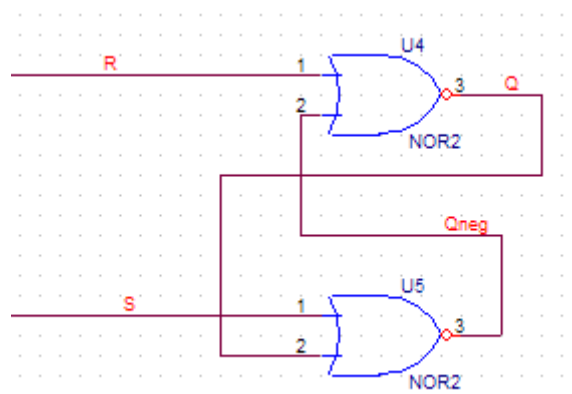
en este caso con componentes que si que se pueden simular dentro del programa de CAD.

Como se puede observar, los interruptores están recreados por puertas de transmisión, para que el circuito funcionara de la forma más perfecta posible para una simulación más clara, la resistencia cuando la puerta de transmisión está abierta es de un valor extremadamente elevado para simular un circuito completamente abierto, en el caso en el que la puerta se cierra, la resistencia de ON es de 1 ohmio, intentando simular un cortocircuito (una resistencia nula) para que así la descarga del condensador fuera lo más escarpada posible hacia tierra, evitando descargas con respecto a una resistencia y por lo tanto exponencial, como ya se ha explicado anteriormente esto no es un comportamiento deseado en el circuito.

El multiplexor en este caso está recreado por dos puertas de transmisión que se complementan en operaciones de encendido y apagado, es decir, cuando una está abierta la otra está cerrada y viceversa, esto es debido a que al comparador sólo le debe llegar una señal para comparar a la vez según los requerimientos del circuito y según el momento del periodo en el éste que se encuentre, esto se consigue simplemente usando un inversor a la entrada de una de las puertas de transmisión que forman el multiplexor, de tal forma que cuando a uno de los interruptores le llegue una señal a nivel alto para que se cierre, al otro le llegará una señal a nivel bajo para que se abra, para ello se complementa la señal de control Cmux según convenga.

El comparador está implementado de la misma forma que en el circuito eléctrico definitivo realizado para el layout, debido a que el modelo de comparador LM311 se encontraba perfectamente definido en las librerías proporcionadas por el sistema de diseño asistido por ordenador OrCAD.

La parte más difícil de entender es cómo simular las señales que activan y desactivan los interruptores sin el uso de un microprocesador, es decir, como se han realizado los biestables RS. En el circuito realizado para la simulación, los biestables RS están formados por puertas NOR como se puede ver en el siguiente esquema.



En el esquema simulado, la salida de un biestable RS se pone a nivel alto en cuanto le llega un pulso a la entrada S (Set) del propio biestable, estos pulsos de entrada son simulados con fuentes de pulsos, cerrando de esta forma el interruptor de carga del condensador en el momento del periodo que se crea conveniente. La acción de apagado del biestable la hace entonces el propio comparador, cuando la tensión en uno de los

condensadores llega al mismo nivel del condensador al que quieren copiar la tensión, la salida del comparador cambia de nivel bajo a nivel alto, activando la entrada R (Reset), del biestable y abriendo el interruptor, pues la salida del biestable sería nula.

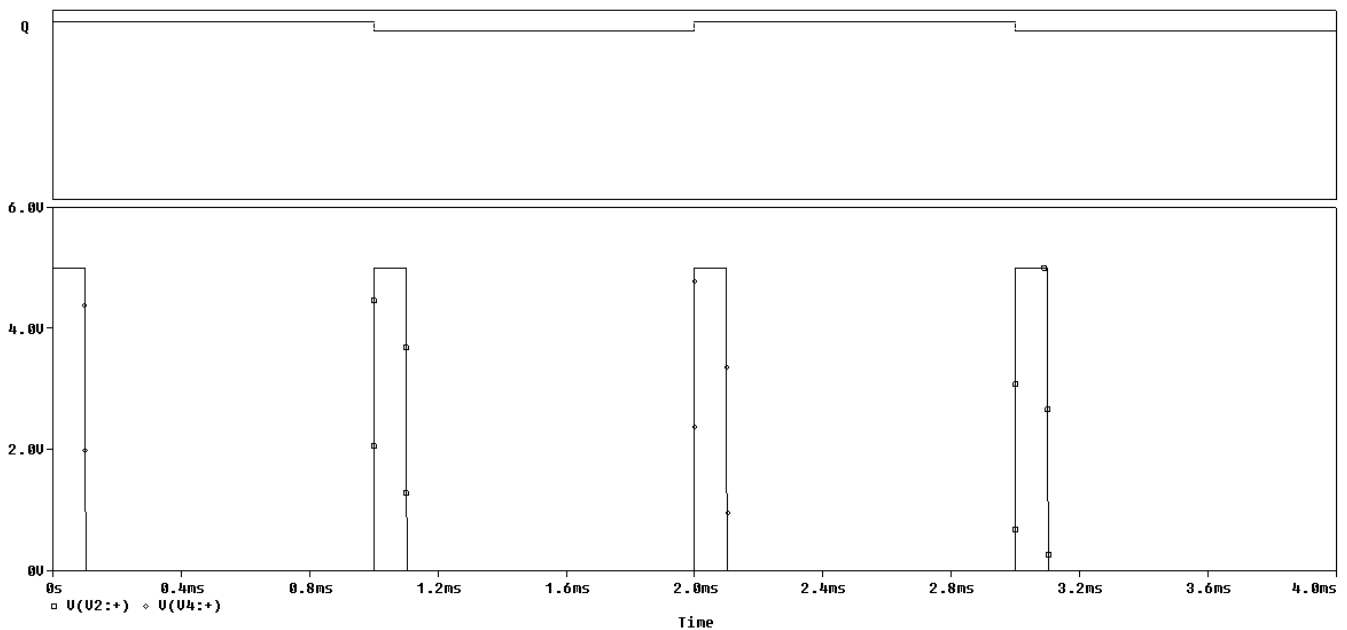


Figura 25: Funcionamiento del biestable RS del circuito.

En la anterior gráfica se puede observar de una forma clara el funcionamiento del biestable RS, la señal V4 (marcada con un rombo) es la señal de control que manda al biestable cambiar su salida a nivel alto, entonces se puede observar como la salida Q se pone a este nivel cuando recibe el pulso de V4, en cambio, cuando al biestable le llega el pulso de V2 en la entrada R, la salida cambia a nivel bajo ya que le ha sido ordenado que haga un reset.

3.1.2.- Esquema eléctrico real del circuito para su fabricación (Figura 30)

Este esquema se corresponde con el diagrama de bloques del circuito construido del apartado 2.3.2. Como se puede ver en el mismo, los símbolos eléctricos no están representados como lo estarían para su simulación, en este esquema están representados los encapsulados de cada uno de los circuitos explicados en el capítulo anterior. Como se ha explicado antes es debido a la falta de los componentes escogidos en el software de diseño OrCAD, para solventar este problema se usaron los conectores predefinidos por el programa que simularían los encapsulados. El problema que esto acarrea es que lógicamente al usar de componentes conectores genéricos, era imposible saber que patilla correspondía a qué función para así hacer el rutado, la solución fue el uso de las hojas de características de los componentes, en las cuales vienen interpretadas la función de cada una de las conexiones del mismo. El uso de estas hojas también se extrapoló a la hora de elegir el encapsulado para su layout, ya que éste tampoco venía lógicamente con el conector genérico.

Los conectores marcados como 4066 son los circuitos utilizados para los interruptores, en el esquema eléctrico pueden verse tanto las patillas usadas para interruptores, numeradas de 1 a 4, como las patillas de control de C1 a C4 que controlan el cierre y apertura de cada uno de sus interruptores.

Para implementar los interruptores se usaron 3 encapsulados, uno para cada uno de los circuitos de carga RC, debido a que así habría una mayor facilidad a la hora del rutado y la mejora en la organización del circuito. Las señales de control de entrada del integrado, como se puede ver en el esquema, vienen directamente de los puertos de entrada/salida que tiene el microprocesador, según se requiera su uso en el periodo del circuito.

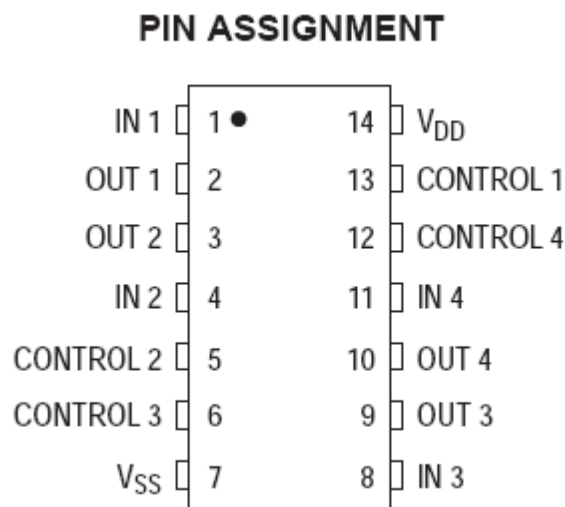


Figura 26: Esquema del encapsulado del integrado 4066.

El conector marcado como 4053 es el multiplexor, según las hojas de características, para que el circuito funcione, el enable debe de estar a 0 por lo que se conecta a tierra, luego tendríamos las conexiones Vee y Vss que son las referencias para los componentes analógicos y digitales internos que tiene el circuito, las conectamos las 2 a tierra ya que queremos que sean iguales. Este dispositivo tiene más multiplexores de los necesarios pero su funcionamiento y los encapsulados serían los mismos que con otro tipo de multiplexores. Por lo tanto no hay problema en que sobren.

Las señales A, B y C, son las que marcan qué salidas están activas y conectadas con qué entrada, según la tabla de verdad proporcionada por la hoja de características (Anexos), en el caso de nuestro esquema, al sólo usar un multiplexor de los posibles, con una única señal de control sería suficiente, pudiendo conectar las otras 2 a tierra al no ser necesarias, y evitando así una posible fuente de ruido a la entrada del circuito.

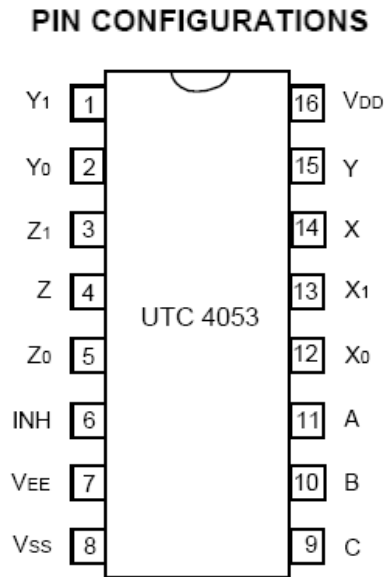


Figura 27: Esquema del encapsulado del integrado 4053.

En el caso del comparador de alta velocidad no hizo falta el uso un conector acorde con el encapsulado proporcionado por su hoja de características ya que el programa si tenía la librería del componente.

Por último el conector marcado como 74HC04 es el que se refiere a los inversores, OrCAD si que puede simular este componente, pero al ser digital, en su símbolo para esquemático no están provistas las conexiones de alimentación y tierra para poder rutarlo, además, al necesitar el demostrador sólo el uso de uno de los 6 inversores contenidos en el encapsulado, hay que hacer que las demás entradas estén conectadas a tierra para evitar así la entrada de posible ruido al circuito. Debido a estos motivos, en el esquema eléctrico se puede ver el encapsulado correspondiente al circuito 74HC04 como el de la figura 28 en lugar al símbolo eléctrico del inversor.

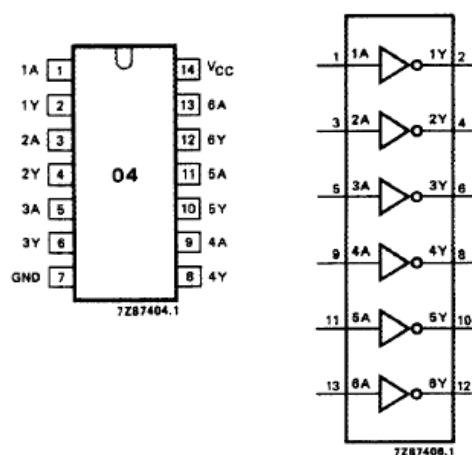


Figura 28: Esquema del encapsulado del integrado 74HC04.

En cuanto al microprocesador hubo que tener en cuenta principalmente su número de conexiones, en el caso del C8051F330 serían 21. Para su rutado eléctrico se usó un conector genérico que nos proporciona OrCAD que contiene el mismo número de estas conexiones que el microprocesador.

El problema era que para su implementación y rutado era necesario entonces crear la huella del componente, ya que al ser un componente no genérico no existía ninguna huella predeterminada para él dentro de la base de datos del programa. Para ello se utilizó directamente el toolstick físico y un calibre para medir todas las distancias tanto para los bordes como la distancia entre pines etc. Una vez obtenidas todas las distancias se utilizó la herramienta Layout Plus (también incluida en el paquete OrCAD) para crear la huella definitiva para su final implementación dentro del circuito y por lo tanto en la PCB final. De ahí y gracias tanto a sus fotos, como hojas de características y la huella para el Layout se pudo hacer perfectamente el rutado tanto de los puertos de entrada/salida como de los puertos de interrupciones externas, ya que era simplemente unir el puerto que se deseara (y el cual viene perfectamente visible junto con su pin en el toolstick), con su número correspondiente en la huella.

Como dato característico se puede observar que la alimentación del microprocesador es distinta a la del circuito, esto es debido a que el microprocesador ha de funcionar específicamente a 3.3V por su tecnología.

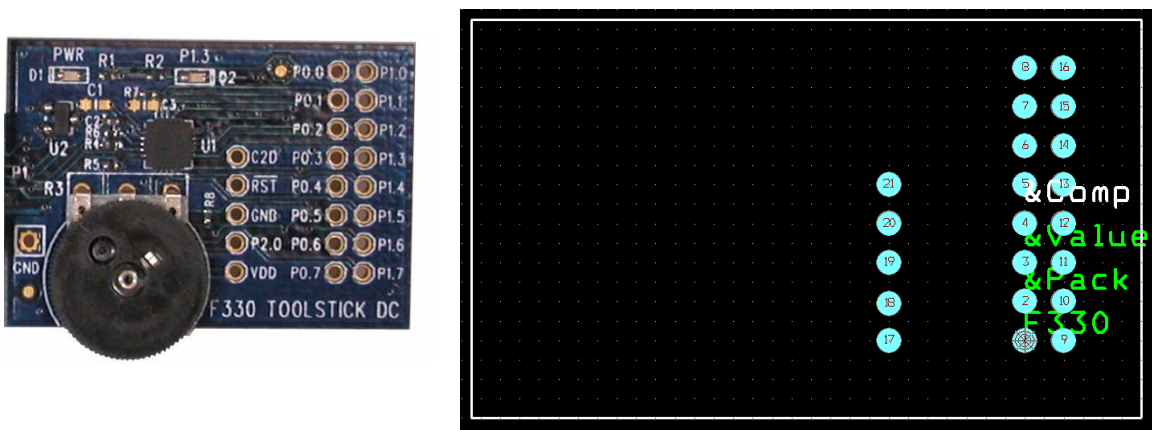


Figura 29: Foto del toolstick del microprocesador y su huella para el layout.

Una vez realizado el rutado eléctrico y comprobado que es correcto, el siguiente paso para la realización del demostrador es realizar el Layout del circuito, que será explicado en el capítulo 5 del proyecto, pero primero se mostrarán las simulaciones del circuito en OrCAD para dar una idea de cómo funciona en distintas condiciones de trabajo.

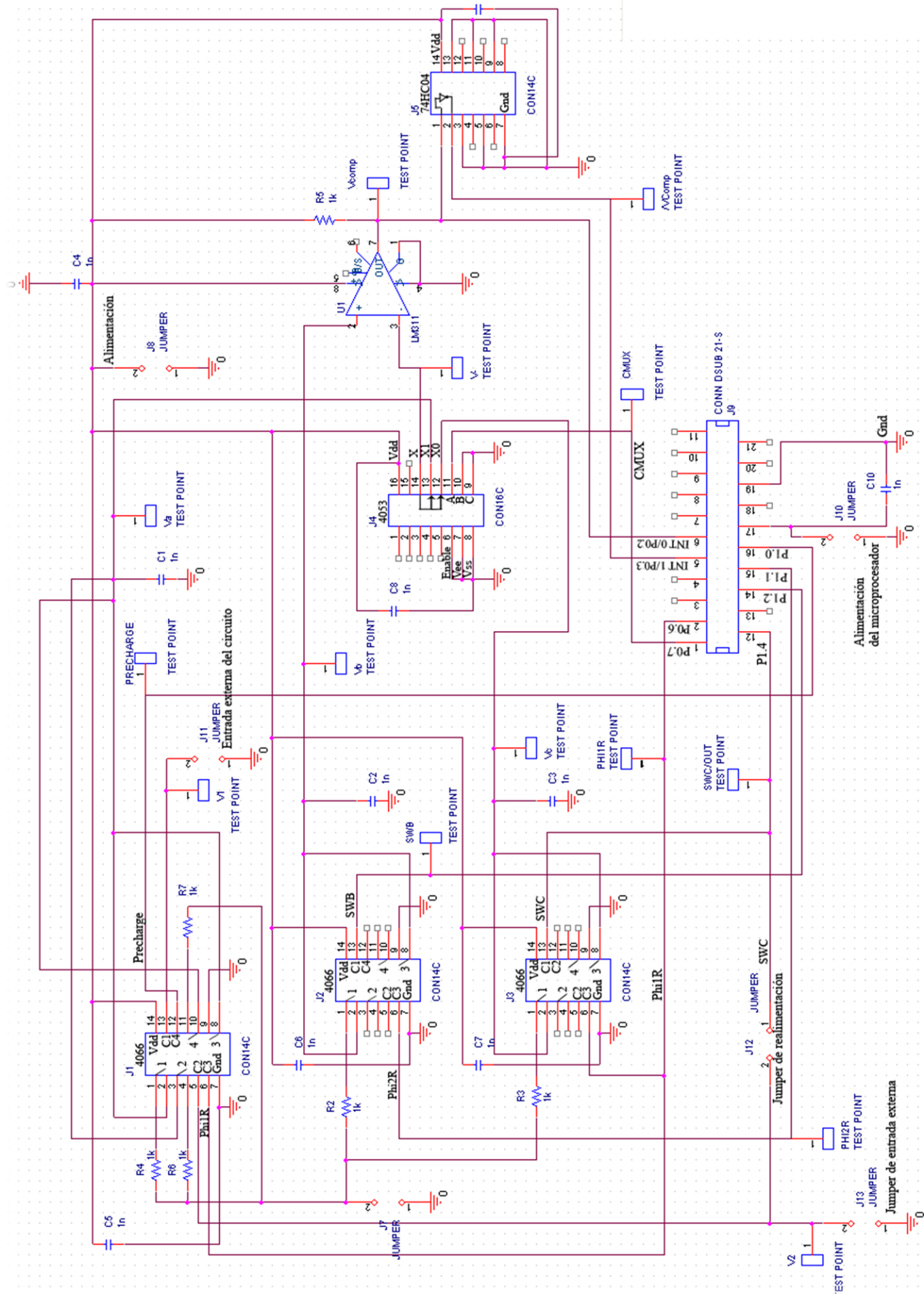


Figura 30: Esquema eléctrico para la fabricación del circuito.

3.2.- Simulaciones.

En este apartado se podrá observar el funcionamiento del circuito en un simulador (concretamente el PSpice que viene dentro del paquete de Cadence OrCAD) para comprobar si se comporta como se esperaba teóricamente, y además probarlo en distintas configuraciones de trabajo.

Se presentarán distintas situaciones de funcionamiento del circuito tanto funcionando como célula de retraso como filtro de paso bajo, y cómo funcionaría según distintas entradas y distintos coeficientes de p .

Funcionamiento del circuito como configuración de retraso:

Esta simulación se realizó con la configuración de la figura 31, como se puede observar, no hay realimentación del circuito para que éste funcione como integrador, o filtro de 1^{er} orden paso bajo, ésta sería la configuración como un bloque en lazo abierto, simplemente un circuito para retrasar la señal 1 periodo:

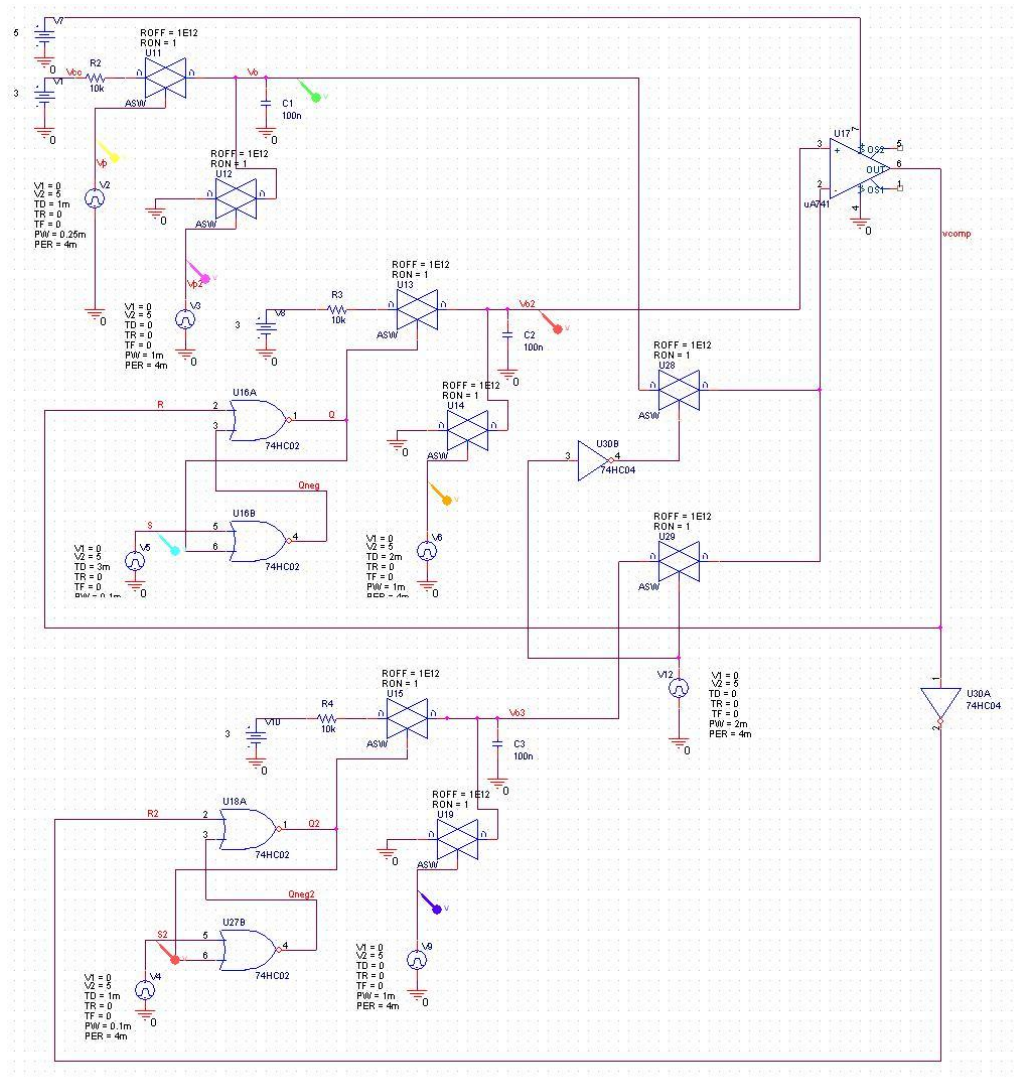


Figura 31: Esquema eléctrico para la simulación de la célula de retraso.

La simulación en régimen transitorio sería:

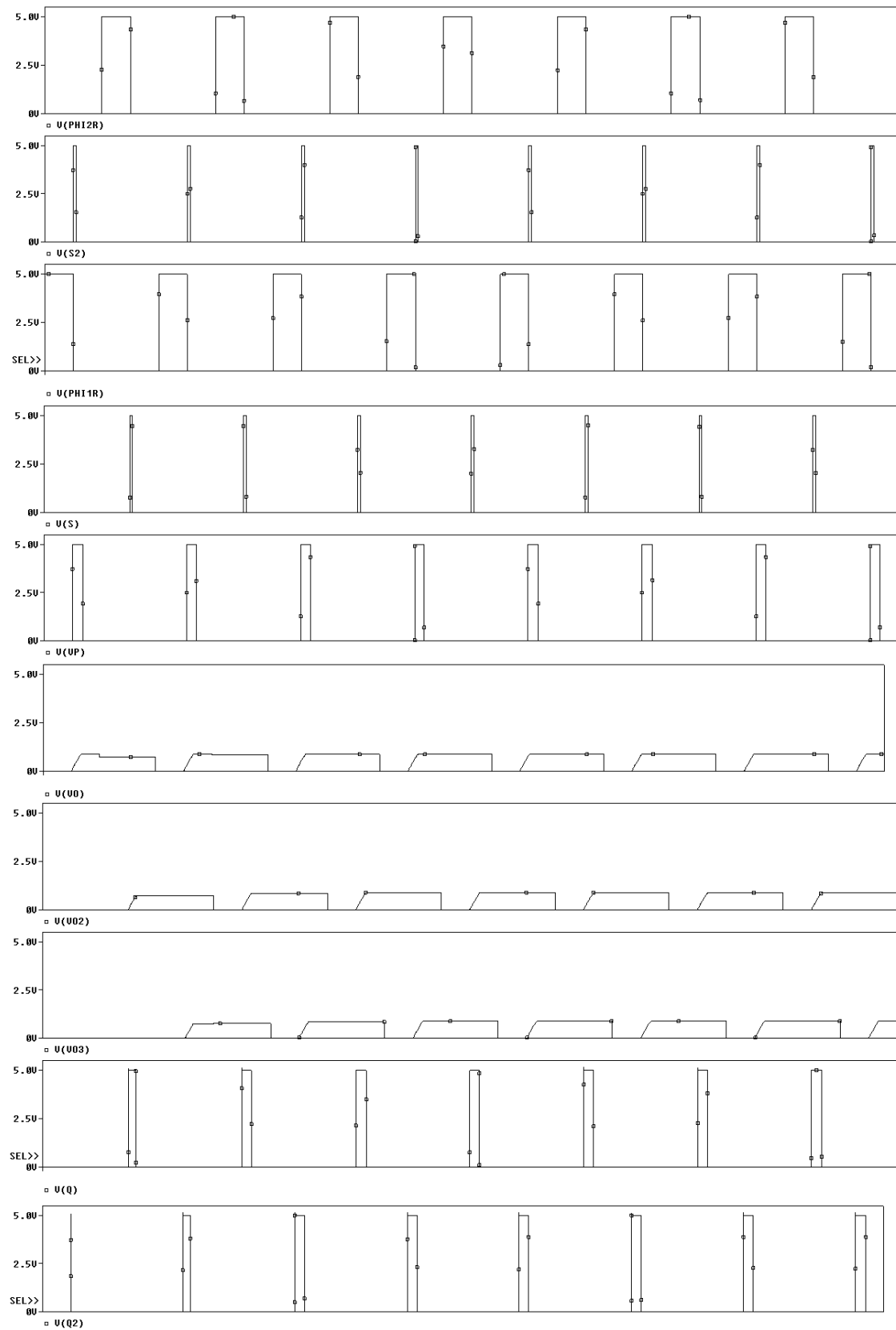


Figura 32: Simulación de la célula de retraso.

En esta primera simulación se representan todas las señales que tienen alguna función importante en el funcionamiento del circuito.

El primer pulso que se da a la entrada del circuito es el pulso VP. En la última gráfica de la simulación, como se puede observar, justamente en el siguiente periodo se da el primer pulso de V(Q2), es decir, esta sería la copia del pulso dado por VP anteriormente. Si nos fijamos se puede ver perfectamente la secuencia de funcionamiento: Primero Phi1R borra la carga en el condensador C_1 , tras esto, VP da un pulso con el cual éste comienza a cargarse, sólo unos momentos después de que se haya cargado, Phi2R borra cualquier posible carga que tenga C_2 , para instantes después dar el pulso V(S) que activa el biestable que carga a este condensador, haciendo la copia intermedia del pulso VP (VQ). En el siguiente periodo, se ve como en un principio C_1 comienza a cargarse de otra vez con un nuevo pulso de VP mientras el condensador C_3 comienza a cargarse también gracias al pulso V(S2) copiando así la entrada que hubo en el periodo anterior. V(Q2).

En las demás gráficas, se pueden observar las señales de control del circuito así como las tensiones de carga de los condensadores, cada una tiene un propósito a cumplir dentro del circuito.

V (Phi1R): Descarga el primer y tercer condensador, nada más comenzar el periodo, se han de descargar los condensadores para evitar así que haya cargas residuales que introduzcan algún tipo de error en el circuito.

V (S): Esta señal activa el biestable RS1, es decir, el que manda copiar la señal de C_1 en la mitad del periodo en C_2 , sería el comienzo de la señal SW_b del circuito.

V (Phi2R): Se encarga de descargar el segundo condensador al comenzar su ciclo dentro de un periodo.

V (S2): Se encarga de iniciar la carga del condensador C_3 , es decir, activa el biestable RS2, para crear la señal de retraso del circuito, concordaría con la señal SW_c en el cronograma.

Simulaciones del circuito como filtro de primer orden (Esquema utilizado igual al de la figura 24).

En este circuito, como se puede observar en la figura 24, la realimentación está implementada mediante el interruptor U34 cerrando el circuito R_5C_1 .

Por defecto, en todas las simulaciones la ganancia de la célula de retraso será 1, modificando únicamente la posición del polo y viendo las consecuencias al modificarlo.

Simulación 2: funcionamiento del circuito con una entrada de tren de pulsos.

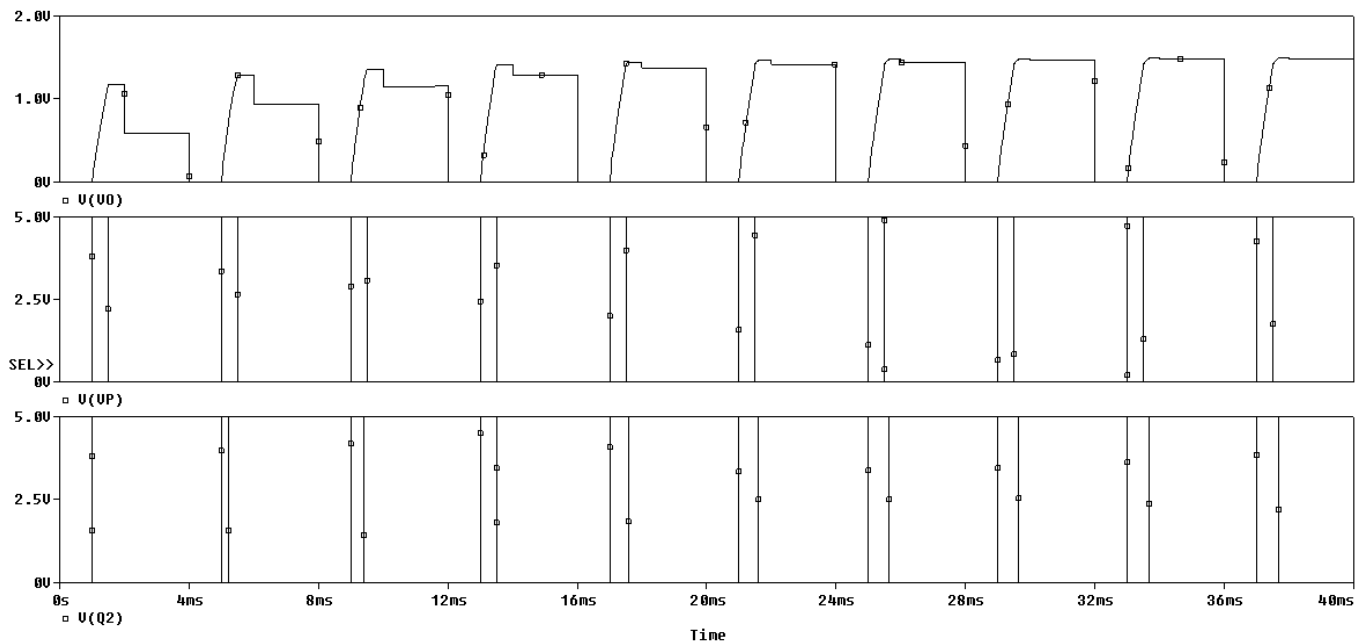


Figura 33.

En esta configuración, las señales de control cumplen la misma función que para el circuito funcionado como célula de retraso, el único cambio realizado en el circuito es la realimentación, pero ese pequeño cambio hace que el circuito tenga una funcionalidad completamente distinta a la anterior que únicamente se dedicaba a retrasar un pulso un periodo, aquí, la señal de salida $V(Q2)$ no es una simple copia, sino que va creciendo paulatinamente e integrándose. Al introducir al circuito un tren de pulsos, el filtro paso bajo actuará integrando la salida exponencialmente, más poco a poco según sea el ancho de entrada del pulso, si el pulso de entrada es muy estrecho, la salida irá aumentando más lentamente que si el pulso de entrada es más ancho, ya que lógicamente al sumar señales más anchas, se llega al máximo del ancho de pulso más rápidamente. Pero la integración de la señal no sólo se verá modificado por el ancho de la onda de entrada, la velocidad de aumento de la misma también estará restringida a los valores que tengamos de los coeficientes p y g como se ha explicado anteriormente. Por ejemplo, en la siguiente simulación veremos como al modificar el valor de p , la salida crece mucho más rápidamente hacia su valor máximo que en esta primera captura. Para modificar la actuación del circuito mediante p hay que hacer que se modifique la resistencia R_5 que cambiará el valor de este coeficiente según se ha visto en las ecuaciones [Ec.10] y [Ec.12]. El motivo por el que únicamente se modifica el coeficiente p , como se ha comentado, es debido a que el coeficiente g es constante de valor 1 en el demostrador presentado para que la célula de retraso haga copias exactas de la señal de entrada y no modifique con una ganancia la onda de entrada original,

Simulación 3: Funcionamiento del circuito con una entrada de tren de pulsos, habiendo modificado el coeficiente p .

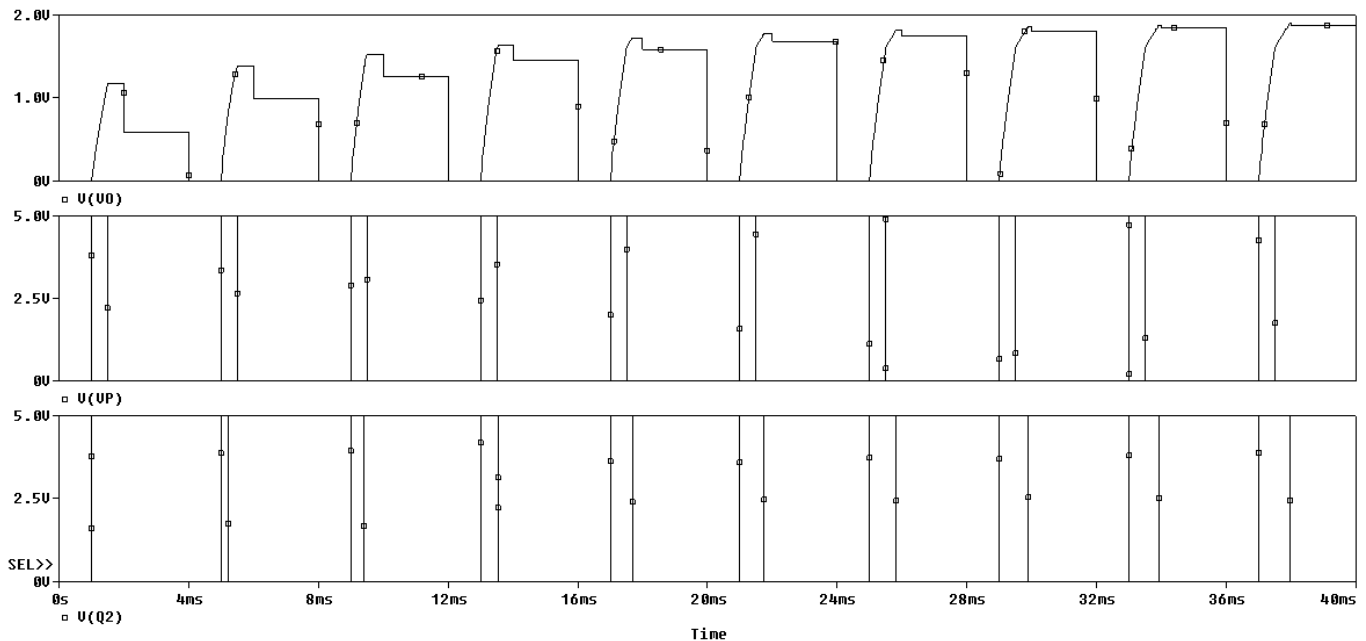


Figura 34.

Para este caso, se ha modificado el valor de R_5 para dar más velocidad a la integración de la onda, para ello este valor de R_5 deberá ser menor al anterior, ya que según la [Ec.10], cuanto menor sea el coeficiente τ_{A2} más elevado será el valor de la señal de salida. Por lo tanto una disminución de R_5 haría que el polo se acercara al eje, obteniendo una mayor rapidez en la integración de la señal.

A diferencia del caso anterior, se puede ver como el crecimiento del ancho del pulso de salida es mucho más rápido, es decir, llega al máximo mucho antes mientras que en el ejemplo anterior no era capaz de alcanzarlo en el tiempo de simulación configurado.

Este máximo está impuesto a 1ms para este circuito con las constantes de tiempo y duración de las fases que se han definido. Es 1ms debido a que es lo máximo que las señales de control dejan cargar a los condensadores, el primer condensador puede cargarse hasta una tensión máxima a la que esté alimentado y seguir alimentándose hasta que se borre su carga mediante Φ_{1R} , pero a la hora de que los demás condensadores copien la señal, para ellos el máximo será $\frac{1}{4}$ del periodo total. Lo que limita la carga de éstos es la señal de control C_{mux} , que intercambia las posiciones del multiplexor, C_{mux} comienza en el 0ms y se alarga hasta 2ms antes de permutar su valor de 2ms a 4ms, si la señal de entrada al circuito es mayor que 1ms, lo que ocurrirá es que cuando se comience a cargar el condensador C_2 a partir de la posición de 3ms de cada periodo (hay que recordar que el tiempo de 2ms a 3ms se invierte en la descarga del condensador), éste comience a cargarse, pero el problema es que el comparador no llega a hacer ninguna comparación al no alcanzar la carga de C_2 nunca a la de C_1 , entonces, en el cuarto milisegundo del periodo, al permutar $CMUX$, la salida del comparador también permutará debido a que en ese instante al comparador le llegaría la tensión

$VC_3(t)$ en lugar de $VC_1(t)$, la cual lógicamente estará a 0V al venir de una descarga, haciendo entonces creer a los biestables que se la comparación se ha realizado. Este es el motivo por el que el máximo ancho de pulso que puede copiar el circuito usado para simular es 1ms. Con el condensador C_3 no haría falta explicar tal proceso ya que copia directamente la señal que hay en C_2 , por lo tanto su ancho de pulso tampoco podrá ser nunca superior al de éste.

Simulación 4: Comportamiento del circuito cuando a su entrada introducimos una entrada escalón.

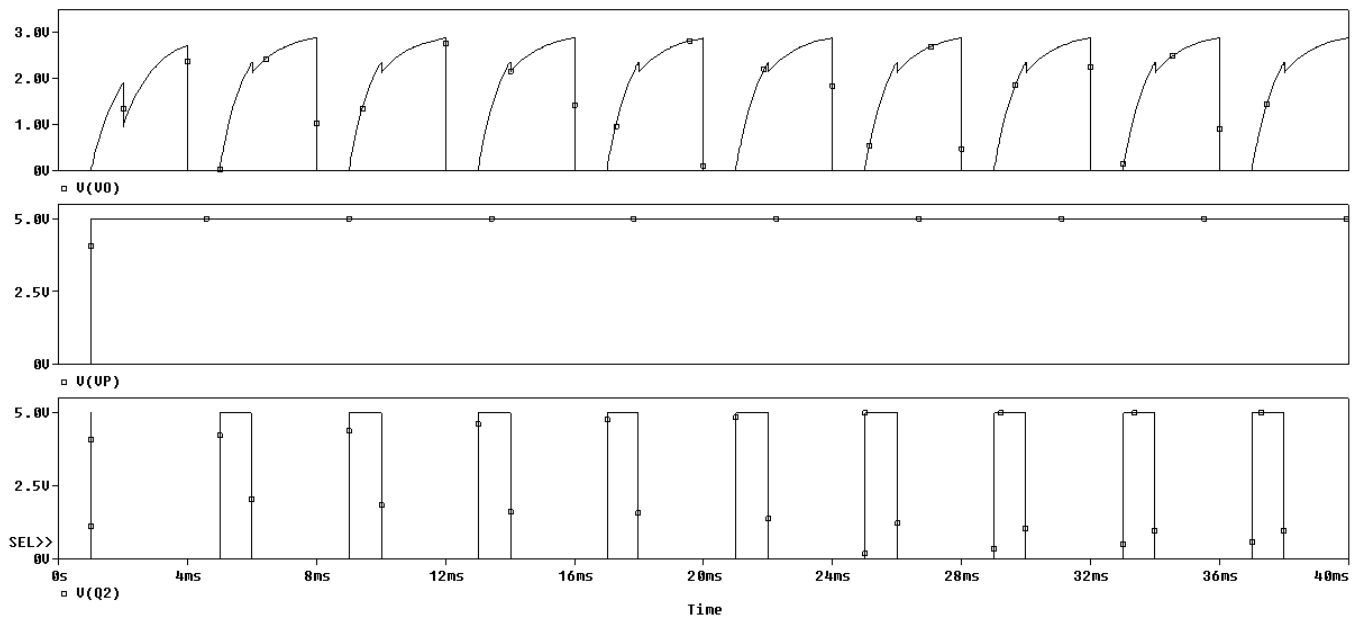


Figura 35.

El resultado de este caso es claramente esperado, puesto que para el circuito una entrada escalón es como un tren de pulsos de ancho de pulso superior a 1ms, por lo tanto al circuito no le da tiempo a integrar la señal antes de llegar al máximo ancho posible que la onda puede tener, ya que si la primera carga del condensador es la máxima posible, la primera copia también tendrá el pulso máximo, impidiendo así que el circuito integre progresivamente la onda.

Simulación 5: Funcionamiento como integrador al introducir y eliminar entrada.

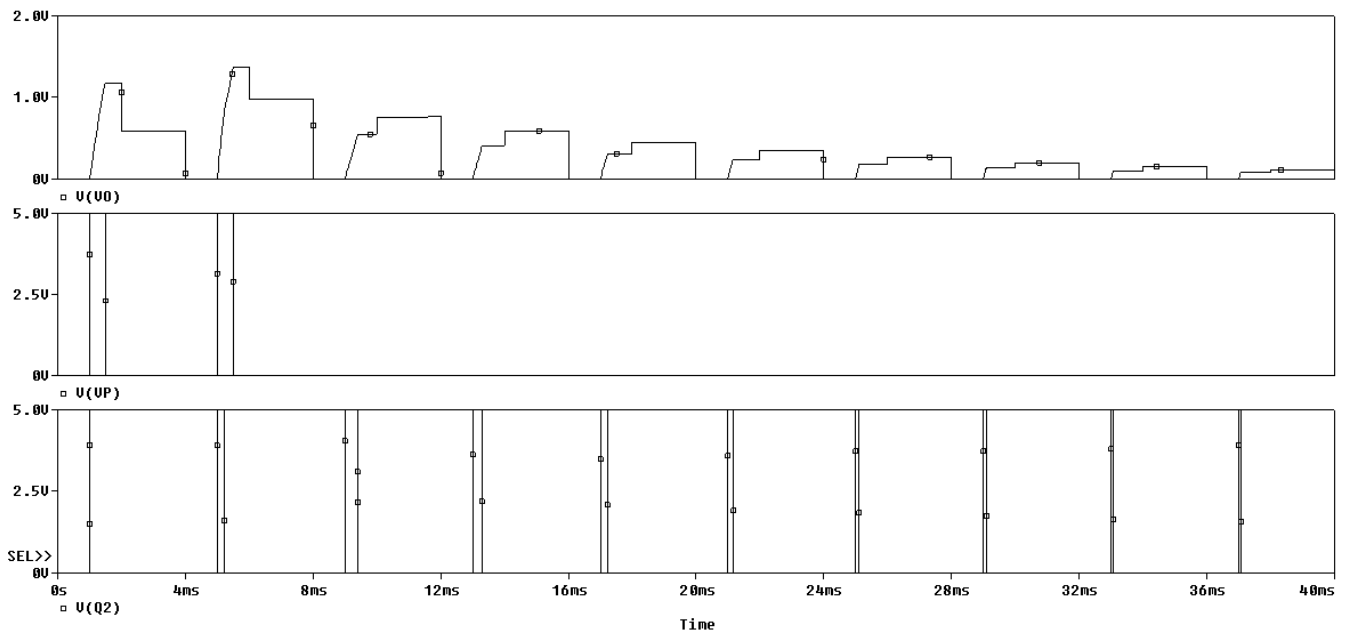


Figura 36.

Tal como el circuito actúa haciendo el ancho del pulso a su salida paulatinamente más grande si se le introduce un tren de pulsos, si el circuito tiene una carga y se anula la señal de entrada, el ancho del pulso de su salida teóricamente debería decrecer poco a poco según el paso de los periodos, ya que esto sería otra forma de integración de la onda. Esto mismo es precisamente el resultado que se obtiene en esta simulación. En un primer momento se introducen dos pulsos de ancho medio al circuito para que los condensadores adquieran una cierta carga, y después se deja al circuito sin más entrada que la que la realimentación le proporciona. Al tener sólo la entrada de realimentación, el primer circuito RC se vuelve bastante más lento (si nos fijamos justo lo contrario que pasaba cuando se sumaban las dos entradas), pues la resistencia de realimentación será mayor que la resistencia del circuito RC para el condensador C_3 , esto es debido a que para que el circuito funcione de forma estable con p menor o igual que 1, el coeficiente τ_{A2} deberá ser mayor que el coeficiente τ_c según se muestra en [Ec.12]. Por lo tanto, al cargarse C_1 sólo con su realimentación, cada vez se cargará a menos tensión y así el pulso resultante será más estrecho con el paso de los periodos. Así pues, como para que el pulso crezca más o menos deprisa se puede modificar el coeficiente p , para que decrezca también a una mayor o menor velocidad se puede modificar este coeficiente, ya que actúa igual en ambos casos.

Simulación 6: Simulación del circuito al introducir una entrada escalón y luego eliminarla.

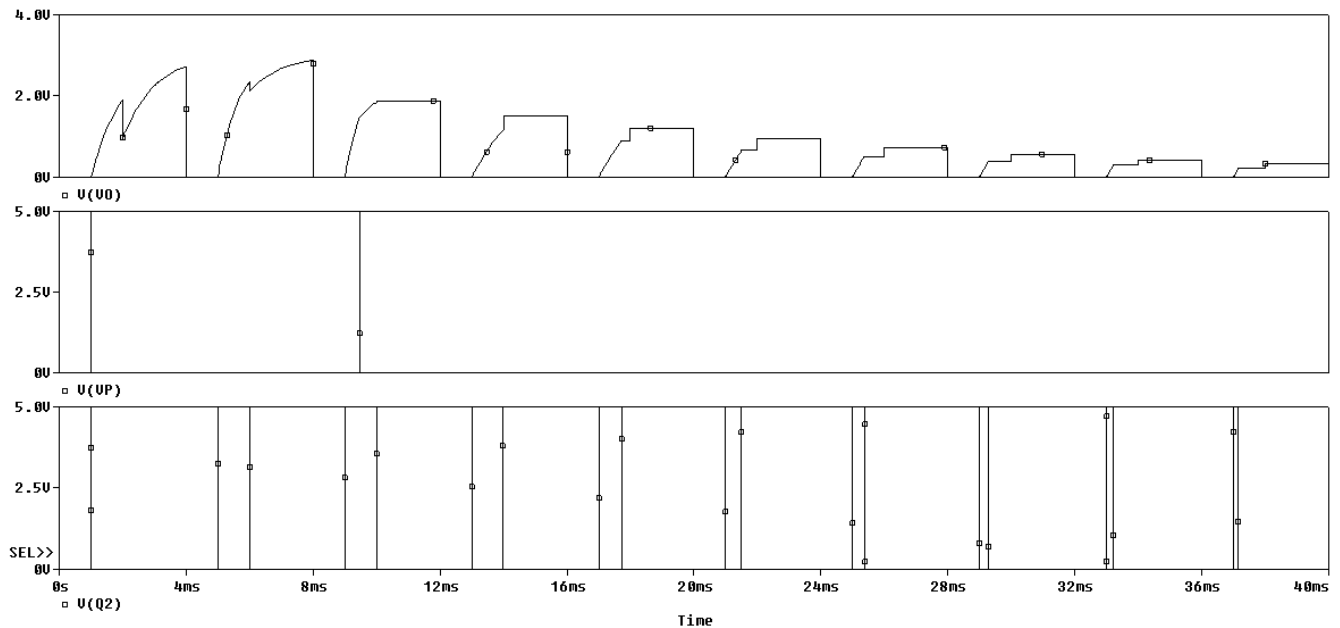


Figura 37.

Esta simulación se basa en una combinación de las dos anteriores para ver el comportamiento del circuito en un funcionamiento alterno. En un principio, al introducir la entrada escalón la salida del circuito pasa a estar directamente en su tope debido a la máxima entrada que se introduce en el circuito. Tras esto, al pasar la entrada de nuevo a 0V, la salida del circuito se va reduciendo paulatinamente como se vio en la simulación 5. Como antes, la velocidad a la que el circuito integra puede ser modificada, pero esta simulación es muy interesante para ver como el circuito es capaz de funcionar perfectamente da igual que tipo de señal introduzcamos, como se adapta a los cambios.

Simulación 7: Simulación del circuito al introducir una entrada escalón y luego eliminarla incrementado la velocidad con p.

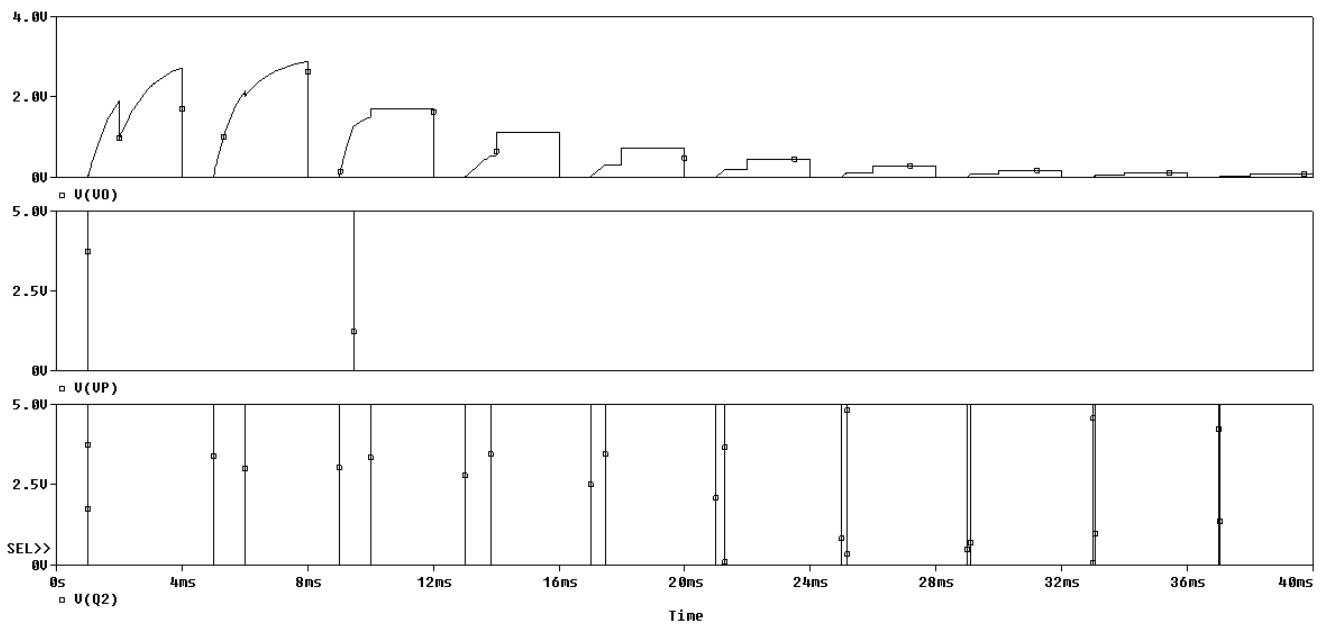


Figura 38.

En esta simulación el valor del coeficiente p se ha modificado para que la integración sea menos suave y en cambio más rápida y extrema, se puede ver sobre todo en el estrechamiento de la onda, donde en los últimos periodos es prácticamente un pequeño tono. En la simulación se observa como el decrecimiento se realiza mucho más rápidamente que en el caso anterior.

3.4.- Cálculo de R y C y su correspondencia con p y g.

Como anteriormente se ha explicado en el documento, hay una correspondencia directa entre el valor de los componentes resistivos y capacitivos que hay en el circuito con los coeficientes p y g de la función de transferencia que representa al mismo.

Tal relación es la siguiente, aunque se ha explicado anteriormente se pondrán aquí los resultados de nuevo.

Constantes de tiempo de los bloques RC:

Para los circuitos RC de entrada, $\tau_{ai} = R_{ai}C_a$

Para el circuito RC del condensador C_3 , $\tau_c = R_cC_c$

Como se comentó en apartados anteriores el condensador 2 no interviene en los coeficientes que aquí se explican.

Por último, los coeficientes p y g vendrán dados por la siguiente relación:

$$g = \frac{\tau_c}{\tau_{A1}} \quad Y \quad p = \frac{\tau_c}{\tau_{A2}}$$

Como la principal misión de este proyecto es crear un demostrador capaz de mostrar un posible funcionamiento de un circuito para su futura implementación en chip, se buscó que se pudiera realizar con componentes comunes de encontrar y a frecuencias fácilmente programables con el microprocesador escogido para el proyecto. Por eso se buscó que el circuito funcionara a frecuencias próximas a KHz ya que para este rango de frecuencia los componentes básicos son fácilmente adquiribles y los retardos que éstos introducen a las señales del circuito son inapreciables, además, este tipo de frecuencias son muy fáciles de programar con el C8051F330 como se demostrará en el apartado de software [4.1.2].

Por lo tanto, el objetivo era conseguir que el circuito funcionara a velocidades de KHz aproximadamente, y este es el motivo por el que los componentes $C_1=C_2=C_3=100n$ y los componentes $R_2=R_3=R_4=10k$, ya que con ellos la constante de tiempo de los circuitos RC será de 1ms, pudiendo así operar entonces a las frecuencias deseadas.

Lo siguiente fue calcular los coeficientes g y p . El coeficiente g , para que la célula de retraso copiara exactamente igual la entrada del circuito, se hizo que fuera de valor 1, es decir que tanto R_2 como C_1 (R_{a1} y C_1 en los apartados teóricos), en el esquemático de la figura 24, valieran exactamente igual que R_4 y C_3 (R_c y C_3). De esta forma, al tener ambos la misma constante de tiempo (se podrían haber cogido otros componentes mientras se mantuvieran las constantes de tiempo, pero por simplicidad se escogieron los mismos) la copia de la señal retrasada será exactamente igual a la copia de la señal de entrada del circuito un periodo anterior, sin introducirle ninguna ganancia.

El coeficiente p no se calculó para mantener un valor constante, se juega con su valor para ver el funcionamiento del circuito y su respuesta. Pero lo que sí se usó para la modificación de este coeficiente fue un potenciómetro en lugar de una resistencia para así variar la constante de tiempo, mientras que el condensador no se modificaba, lógicamente porque es el mismo usado por R_2 , es decir, C_1 . Como hemos dicho antes, la variación del valor p es limitada, ya que el funcionamiento del circuito depende directamente de su valor, y si éste valor p es mayor que 1, el circuito estaría en un punto de trabajo inestable en el que no funcionaría correctamente.



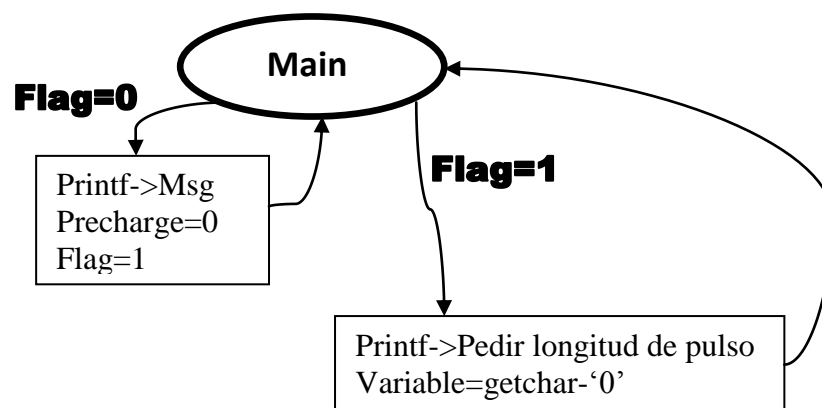
4.- Diseño del Software.

En este apartado quedará completamente explicado cómo se pensó y se programó el software necesario para que el microprocesador hiciera que el sistema funcionara tal y cómo se había definido según los cronogramas teóricos y sus simulaciones mediante OrCAD.

El lenguaje de programación utilizado fue C, evitando así la programación usando lenguaje máquina, simplificando así la tarea de programación del microprocesador y haciéndola mucho más fácil e inteligible.

4.1.- Organigrama del programa.

4.1.1.- Organigrama del programa principal.



La función del programa principal es muy sencilla, en un primer momento, cuando el flag es igual a 0, es decir justo al principio de la ejecución del software, su principal tarea es mostrar un mensaje de bienvenida presentando a la persona que ha realizado el proyecto de fin de carrera, pero lo más importante que se realiza en este inicio del programa cuando la variable flag aún no se ha activado, es poner la salida del puerto precharge a 0, esta es una forma de asegurar que el microprocesador tenga desde un primer momento la salida de este puerto a nivel bajo, haciendo así que el circuito no comience a funcionar fuera de una secuencia de control, lo que podría desembocar en errores y mal funcionamiento ya que la salida del puerto precharge va directamente conectada a uno de los interruptores que cierran el primer circuito RC como se explicó en [2.3.2], que es el que hace que el circuito comience a funcionar.

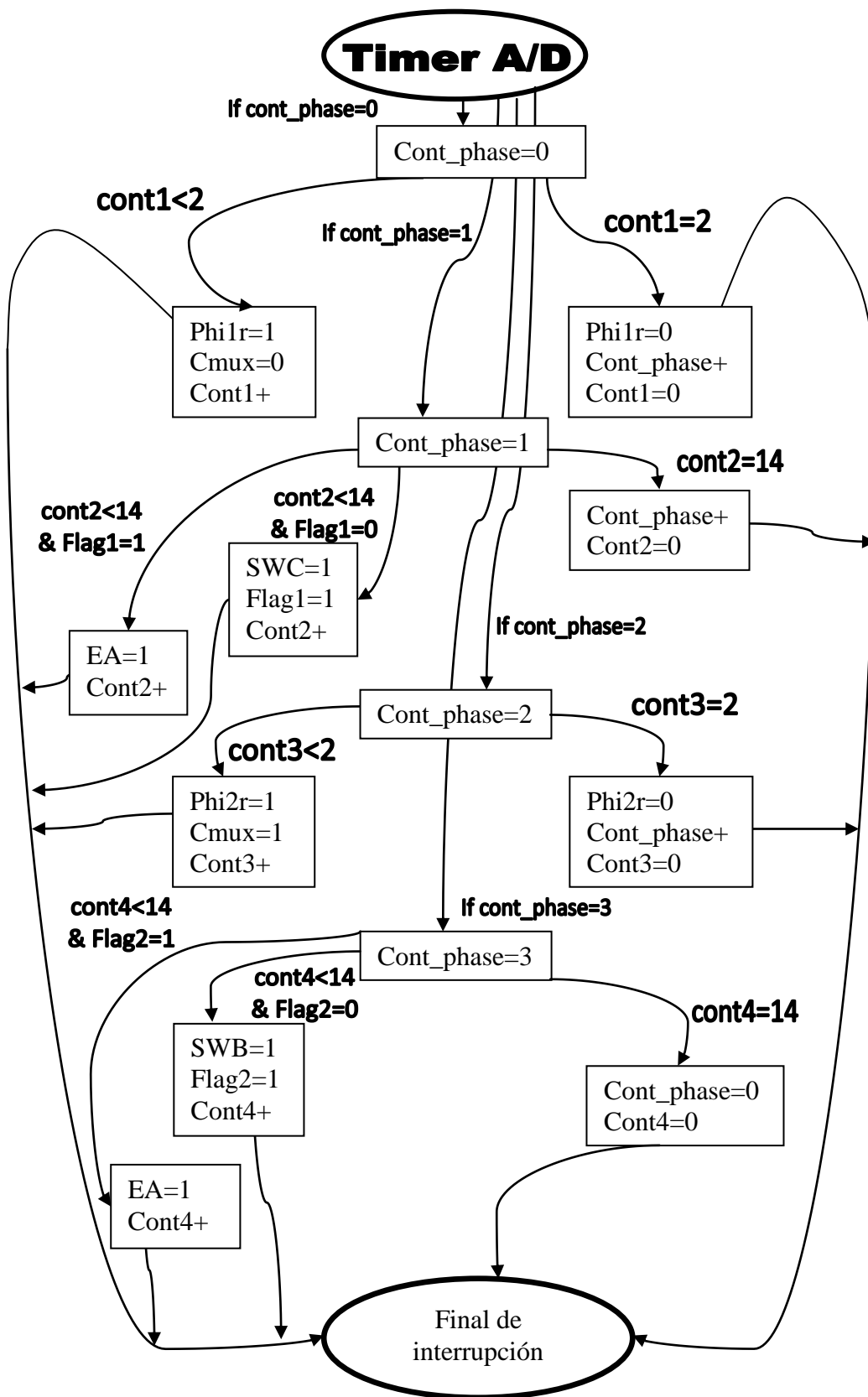
Una vez realizada esta operación, lo siguiente que hace el programa principal, es pedir por pantalla el ancho del pulso que queremos que se introduzca para cargar al condensador C_1 , es decir, el programa pide que se le diga cuánto tiempo va a estar el puerto precharge a nivel alto. El motivo porque el que el demostrador usa el puerto precharge en lugar de entradas externas recordamos que es simplemente por la facilidad con la que se pueden modificar tanto el ancho de pulso como la posición dentro del periodo del ciclo de trabajo del mismo en el circuito en el que el condensador se carga, aunque el circuito demostrador construido también tiene la posibilidad de introducir entradas externas, aunque debido a la dificultad de sincronizarlas con el mismo mediante generadores de funciones corrientes hace que se prefiera el uso de una señal interna del micro.

Como se puede ver, al acabar de pedir el pulso por teclado, el programa sigue funcionando en un bucle infinito, pidiendo de nuevo que se introduzca un valor para el ancho de pulso, de esta forma podemos modificarlo siempre que queramos sin necesidad de reiniciar el programa.

Para introducir un número por teclado se hace un `getchar` para obtener el dato y luego a éste se le resta el carácter '0', esto es debido a que mediante el software proporcionado para trabajar con el micro, los únicos datos que se le pueden enviar son en formato ASCII. La solución para ello se basa en que restando el carácter '0' a un número en ASCII, se consigue que se haga la conversión a número real, es decir, por ejemplo: queremos mandar al microprocesador un número 5, si lo enviamos directamente en realidad lo que se hace es enviar al programa el carácter 5 en ASCII, al cual le corresponde el número 53, éste, como es lógico, no tiene nada que ver con el ancho de pulso que queremos introducir pero es lo que el programa entenderá. Así pues, si sabemos que al carácter 0 le corresponde el número 48 en ASCII, y hacemos $53 - 48$, se obtiene justamente 5, que es lo que finalmente mandará el programa al micro y justo lo que se quiere mandar desde un principio.

Este algoritmo funciona con todos los números entre 0 y 9, por lo tanto hay que tener cuidado de no mandar un número que salga de este intervalo ya que por ejemplo un 55 en ASCII sería 5353 y lógicamente el algoritmo no funcionaría.

4.1.2.- Organigrama de la interrupción del A/D.



El código más importante del programa, que lleva la configuración de todas las señales de control del circuito así como de la señal que se introduce al propio circuito es el de la interrupción del timer A/D, el uso de esta interrupción en lugar de otra es debido a que ésta es la que más veces se dispara por segundo, llegando a un total de 7812 reiteraciones, con las cuales se pueden llegar a frecuencia de trabajo de hasta 4 KHz, y que son muchas más que las que tiene el timer 0 configuradas por defecto, que son aproximadamente 30 iteraciones por segundo. Por lo tanto, y como cada una de las fases del circuito en un principio han de ser de 1ms de ancho según se han definido en el cronograma, su programación es muy factible usando la interrupción de este timer.

El cálculo de las interrupciones necesarias sería el siguiente: Si en un segundo hay 7812 interrupciones, cada interrupción tiene una duración de $1,28 \cdot 10^{-4}$ segundos, por lo tanto, 1ms serán aproximadamente 8 interrupciones. Así pues, un periodo completo, con las 4 fases, serían 32 interrupciones, o lo que es lo mismo, 4 milisegundos.

Pero a la hora de ver el organigrama de este timer vemos como las fases no corresponden en tiempo a lo que hemos visto anteriormente en las simulaciones y cronogramas, pero esto tiene una explicación pues es percibible que, aunque el total de las 4 fases sigue siendo 32 interrupciones, algunas fases únicamente ocupan 2 interrupciones, mientras que otras ocupan 14. Las que ocupan 2 interrupciones son las fases en las que únicamente hay que descargar los condensadores, como esto es un proceso extremadamente rápido, no es necesario un mayor tiempo, y de esta forma, en las fases donde es más crítico controlar el tiempo así como la exactitud de las señales, hay más tiempo para maniobrar y para que las cargas de los condensadores aumenten, pudiendo obtener entonces a la salida pulsos más anchos y comparaciones más exactas que de la otra forma.

La manera de llevar el tiempo y por lo tanto las interrupciones en el programa es mediante contadores, cada fase tiene su propio contador, haciendo que éste vaya sumando de 1 en 1 cada vez que se dispara la interrupción del timer, se consigue llevar el número de interrupciones que se han realizado y por lo tanto controlar el tiempo que el programa se encuentra en una fase del mismo.

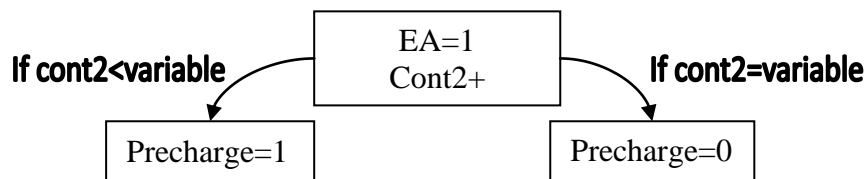
Ahora se va a proceder a explicar paso a paso el funcionamiento del programa en la interrupción del timer A/D.

Cuando la fase es igual a 0, es decir, el programa se encuentra en el primer momento del periodo, lo principal que ha de hacer es eliminar posibles cargas iniciales o cargas anteriores tanto en el condensador C_1 como en el condensador C_3 , de esta forma evitamos errores de cargas residuales y por lo tanto errores en la copia de señales. Para ello, el micro activa la salida del puerto llamado Phi1R, poniéndola a nivel alto, y consiguiendo que se cierre el interruptor que hace que el condensador se descargue.

Además, en este momento la salida del puerto Cmux del microprocesador es puesta a 0, ya que si observamos la figura 30 podríamos decir que a la tensión de C_3 le corresponde la entrada X_0 en el multiplexor, la cual si miramos las hojas de características del mismo se activa con la combinación lógica 000, este es el motivo por el que Cmux tiene que estar a nivel bajo en esta fase.

Una vez realizado toda esta cantidad de tareas, el programa lo que hace es desactivar la señal Φ_{1R} , ya que en este momento los condensadores ya se habrán descargado y lo siguiente que han de hacer es volver a cargarse para copiar señales o recibir una nueva señal de entrada. Para finalizar esta fase, primero se cambia de fase mediante la variable `cont_phase`, y por último se pone el contador que ha llevado el tiempo de esta fase a 0 para que cuando vuelva a repetirse el ciclo periodos esté listo para actuar de nuevo.

En la siguiente fase, la fase número 1, o `cont_phase=1`, es cuando los condensadores C_1 y C_3 se cargan, el primero se carga a una señal nueva en el periodo actual, y el segundo copia la carga que C_1 tenía en el periodo anterior. Como hemos explicado antes, la carga del condensador C_1 se realiza mediante software y un puerto interno en lugar de una señal externa, simplemente para realizar pruebas en el circuito. Para crear la señal de entrada al primer condensador, el programa utiliza la variable definida en el programa principal, a la cual se introduce por teclado el ancho del pulso deseado. Para que el programa funcione correctamente y no de ningún error, este número debería ser menos que 10 para que se pueda realizar la conversión de ASCII a número real. Una vez dicho esto, para hacer entonces la señal precharge del ancho deseado e introducido por teclado simplemente hay que mantenerla a nivel alto hasta que el contador de la fase supere al valor de la variable que se ha guardado, tras esto se pondrá a nivel bajo de nuevo. Es decir se añadiría una pequeña modificación al código de forma que en la fase número 1 se incluiría lo siguiente:



Para hacer el retraso de la señal, en esta misma fase se debe activar el interruptor que cierra el circuito para el condensador C_3 , es decir, se activa la señal SW_c , para ello simplemente se pone a nivel alto. Pero hay que tener mucho cuidado en un concepto, la activación de la señal debe ocupar el menor tiempo posible al programa, es decir, la puesta de SW_c a 1 debe ser lo más rápida posible, esto es muy importante ya que si por ejemplo la señal SW_c se deja durante toda la fase igualada a 1, estaría obligando al puerto a poner su salida a nivel alto durante prácticamente 2ms. Entonces, si la señal que tiene que copiar del periodo anterior es menor a estos 2ms, que es lo más probable, lo que ocurrirá será que al haberse igualado la carga en los condensadores y por lo tanto el comparador haber mandado la alerta de que SW_c debe ponerse a 0, ésta se pondrá a nivel bajo, pero inmediatamente volverá a ponerse a nivel alto en cuanto vuelva a entrar en la interrupción del timer A/D y por lo tanto en la fase 1 de nuevo. Por ello, si la activación del SW_c a 1 se hace en lo que dura una interrupción del timer, no ocurrirá ese problema al circuito, ya que el programa al volver a la fase 1 tras poner a nivel bajo SW_c , no intentará reactivarla. Esto se consigue mediante la variable denominada `flag1`, esta variable sólo estará a 0 la primera vez que se entre a la fase, si se hace que únicamente si se cumple la condición de que `flag1` sea 0 se active la señal SW_c , se acabará con este error.

Como antes, cuando la duración de la fase llegue al máximo de interrupciones definidas para ella, lo que hará el programa será cambiar de fase hacia una posterior, y poner el contador de esta fase a 0 para que esté lista en el próximo periodo.

En la siguiente fase, la fase número 2, el programa ha llegado hacia la mitad del periodo, por lo tanto es el momento para que el condensador 2, que es el que hace la copia intermedia de la señal, actúe. Por ello esta fase 2 será análoga a la fase 0 explicada anteriormente, en este caso la señal de borrado del condensador será Φ_{2R} en lugar de la anterior Φ_{1R} . Además, en esta fase la señal C_{mux} cambia de nivel, esto es debido a que en la siguiente fase lo que el circuito tendrá que hacer es que la carga de C_2 sea exactamente la misma que la de C_1 medio periodo antes. Por lo tanto el comparador tendrá que comparar la tensión $V_{C_2}(t)$ con la de $V_{C_1}(t)$ en lugar de tener que comparar la tensión $V_{C_2}(t)$ con $V_{C_3}(t)$.

Al acabar esta fase, como antes, se desactiva la señal de descarga del condensador ya que éste ya estará completamente descargado, se cambia de fase y se reinicia el contador.

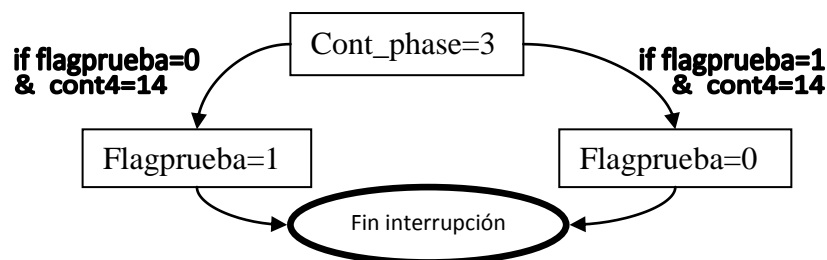
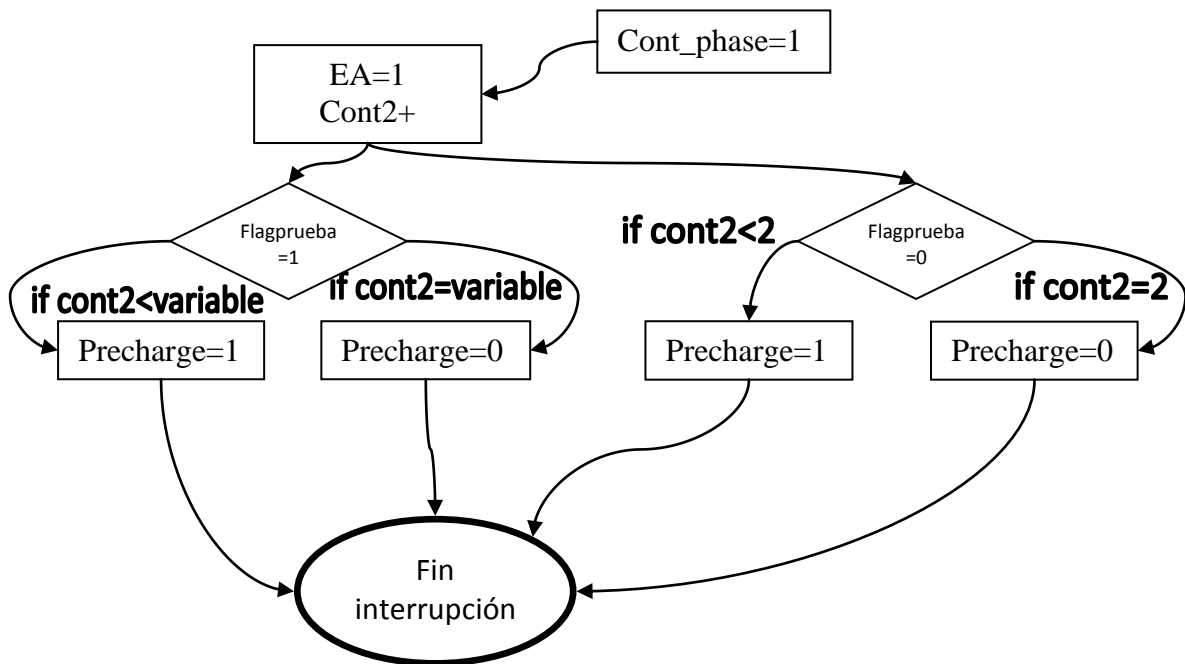
La fase 3 es la análoga a la fase 1, es decir, se encarga de activar la carga del condensador 2. Como en el caso anterior, hay que tener cuidado para que la activación de la señal SW_b sea lo más rápida posible para que no haya errores a la hora de copiar la señal. Por lo tanto, se usará otro flag como en la fase 1, aunque en este caso la variable será flag 2.

Al terminar esta fase, aparte de poner el contador de la fase a 0, se manda al programa a la fase inicial, para que comience el ciclo de nuevo.

4.1.2B.- Modificación del programa de interrupción de Timer A/D para probar célula de retraso.

Como se puede ver en los anexos (9.1.3), hay una pequeña variación del programa de esta interrupción para probar el circuito como célula de retraso en el osciloscopio y poder verla correctamente. La variación consiste en permutar en cada periodo el valor de la señal precharge entre dos posibles, de esta forma, si se ponen la entrada y la salida del circuito en el osciloscopio será mucho más fácil observar cómo se ha retrasado la onda, pues al ser pulsos de ancho diferente, es muy sencillo ver si se han retrasado un periodo ya que si se ve a la salida del circuito un pulso de un ancho definido en un periodo T , y se ve a la entrada ese mismo pulso pero en un periodo $T-1$ se podrá decir que se ha conseguido retrasar la señal.

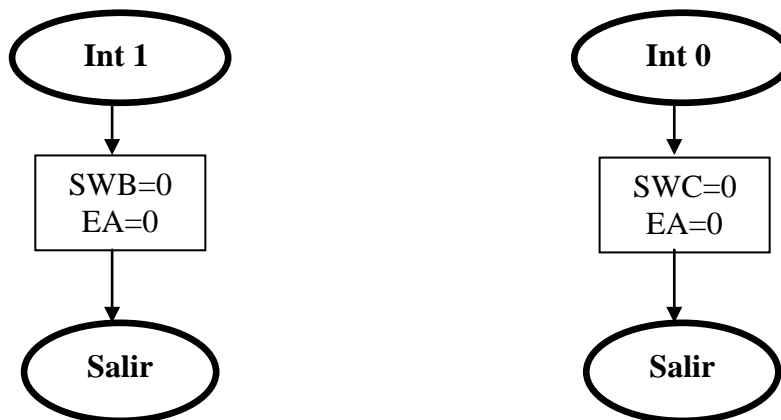
Las modificaciones en el organigrama serían en la fase 1 y en la fase 3, se representarán a continuación.



El funcionamiento de esta modificación es muy simple, en la fase número 1, a la hora de crear la señal de entrada al circuito, el programa pregunta si la variable flagprueba está a 1, si es así la señal percharge que crea el micro es del ancho de pulso que hemos introducido por teclado, y por lo tanto de ancho de pulso que queramos. En cambio, si la variable flagprueba está a 0, el micro lo que hace es que la señal de entrada al circuito precharge tenga un ancho de pulso fijo y de valor 2, de esta forma conseguimos que el micro sea capaz de proporcionar dos anchos de pulso para poder así observar el retraso en el osciloscopio.

En la fase 3, simplemente se permuta el valor de flagprueba cada periodo para así poder hacer lo que hemos explicado anteriormente de ser capaces de introducir dos ancho de pulsos distintos al circuito.

4.1.3.- Organigrama de las interrupciones externas 0 y 1.



La parte de código que reside en las interrupciones externas es muy sencillo. Puesto que la misión de las interrupciones es abrir los interruptores de los circuitos RC cuando éstos están a la tensión deseada, en cada una de los programas de interrupción habrá que poner a nivel bajo la señal que activa dichos interruptores y que serán las copias de los pulsos de entrada. A la interrupción 0 va conectada la señal que indica que el condensador C_3 ha llegado al nivel de carga al que estaba el condensador C_1 en el periodo anterior (Es decir, C_3 ha copiado la carga que tenía C_2). Y a la interrupción 1 va conectada la señal que indica que el condensador C_2 se ha cargado a la misma tensión a la está el condensador C_1 en ese mismo periodo.

El funcionamiento de estas interrupciones como se ha dicho en capítulos anteriores es mediante flancos, cuando se activa una de las señales SW durante la ejecución del programa, los condensadores empiezan a cargarse y al llegar a las tensiones deseadas, el comparador permuta su salida, esta permutación es lo que hace al programa entrar en las rutinas de interrupción, en las cuales desactiva la señal SW correspondiente para que el condensador deje de cargarse. Hay que decir que las rutinas de interrupción externa son las que mayor prioridad tienen dentro del orden de ejecución del programa, por lo que en cuando se recibe un flanco por el puerto de interrupción externa el programa salta directamente a ejecutar esta parte del código.

A parte de esto en los programas de interrupción se puede ver como se desactivan las propias interrupciones externas, esto simplemente es una medida de seguridad para que estas interrupciones no sean disparadas por algún tipo de ruido que pueda producirse en el circuito, por ello aquí se desactivan y son activadas cuando se tiene certeza que se van a disparar debido a las señales recibidas por el comparador LM311.



5.- Construcción y pruebas.

Una vez realizadas todas las simulaciones y viendo que desde los programas de diseño asistido la idea debería funcionar, el siguiente paso es el diseño de la placa PCB del circuito (sus huellas, padstacks etc.) y una vez construida, su montaje y finalización con las pruebas que confirmen el funcionamiento del demostrador de una forma correcta. En este capítulo se mostrará el procedimiento seguido para la realización de todo esto, el paso a paso de cómo se diseñó y construyó la PCB y cómo se realizaron las pruebas en el laboratorio.

5.1.- Diseño de la PCB.

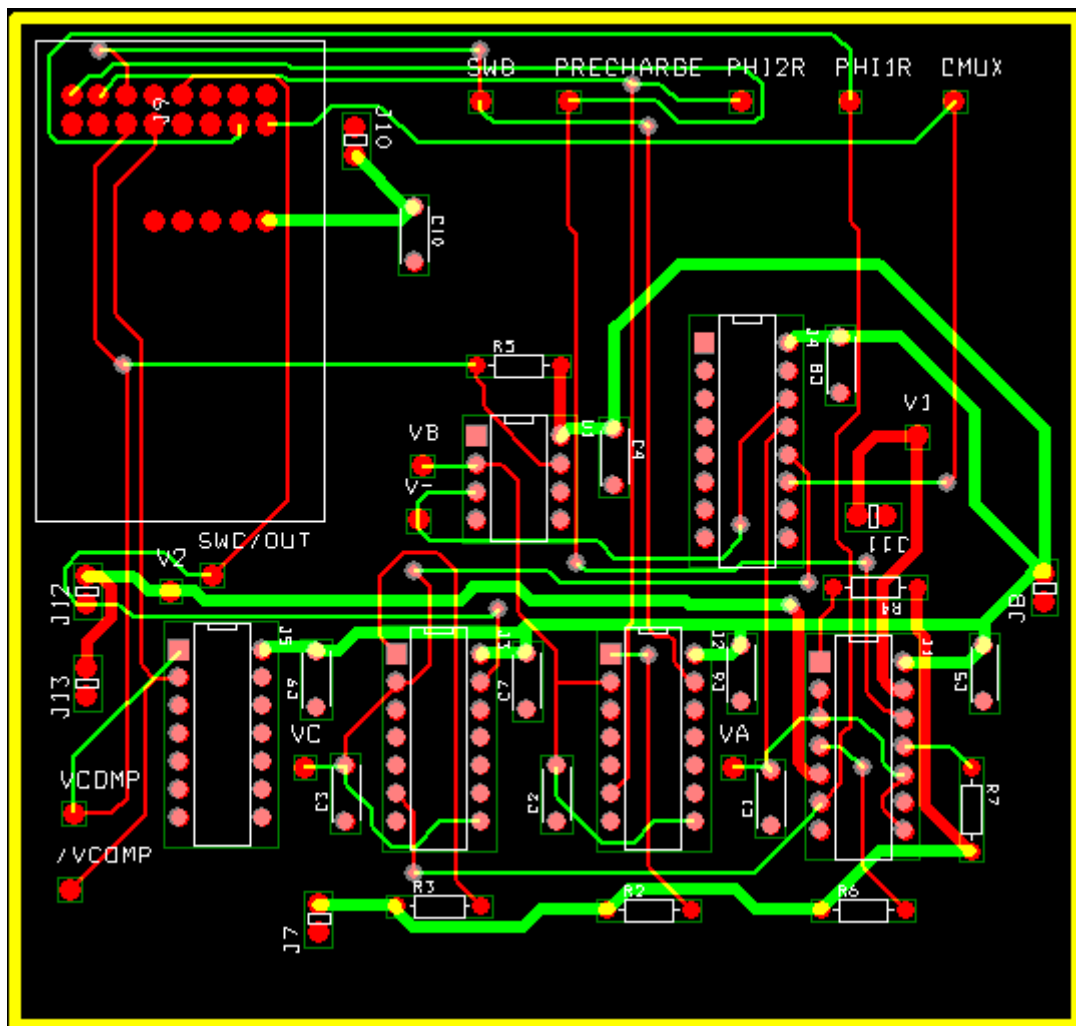


Figura 39: Diseño en OrCAD Layout de la PCB.

La PCB fue diseñada con el programa OrCAD Layout (este programa pertenece a Cadence como los demás programas utilizados para la realización del proyecto), aprovechando la compatibilidad entre el diseñador de esquemáticos y el simulador PSpice con él a la hora de modificar tanto las huellas de los componentes así como el ancho de las rutas y los pads.

Como también es posible ver en el esquemático y en éste layout, cada uno de los integrados así como el microprocesador tiene un condensador de desacoplo en su patilla

de alimentación de forma que se protege al circuito de posibles variaciones en la misma, haciéndolo más estable.

En esta figura 39 se puede ver el rutado definitivo de la placa, las dimensiones de ésta son 3795 por 3615 mils e incluye un plano de masa, aunque la representación del mismo no se ha incluido en la figura ya que si no sería muy complicado ver el rutado de la placa correctamente. Este rutado se realizó mayoritariamente de forma manual (se puede observar por ejemplo en el rutado de las pistas de alimentación) aunque en zonas con una mayor densidad de redes hubo un apoyo de la herramienta de auto rutado que posee el software utilizado para simplificar el trabajo.

5.1.1.- Huellas de los componentes.

Como se ha explicado antes, el OrCAD no tiene información sobre los componentes utilizados para implementar tanto los interruptores como el multiplexor y el microprocesador, por ello se usaron unos conectores genéricos proporcionados por las librerías de OrCAD Capture en el esquemático a los cuales se les añadió la huella exacta que deberían tener esos componentes en la realidad sacándolas de las hojas de características de los mismos. Para los componentes electrónicos que si se encontraban en la base de datos del programa no fue necesario modificar su huella ya que la que venía por defecto era la correcta como por ejemplo la del LM311. Por lo tanto las huellas utilizadas serían:

CD4066B: DIP.100/14/W.300/L.800
CD4053: DIP.100/16/W.300/L.900
74HC04: DIP.100/14/W.300/L.800
LM311: DIP.100/8/W.300/L.450
C8051F330: Huella creada manualmente.

DIP significa que se está usando un encapsulado tipo DIP, .100 significa que la distancia entre los centros de los pines es de 0.1 pulgadas, el siguiente dato es el número de pines que tiene el encapsulado. W.XXX indica el ancho del encapsulado en pulgadas entre los centros de los pines mientras que L.XXX indica la longitud total del encapsulado, como antes, también en pulgadas.

5.1.2.- Ancho de las pistas.

Para la construcción del circuito se usaron dos anchos de pistas distintos, uno para el rutado de sus pistas de alimentación y otro para el rutado de las pistas de conexión.

Para el rutado de las pistas de alimentación, el ancho se fijó hasta los 50mils, de esta forma es posible asegurarse de que no hay ningún problema a la hora de que estas pistas soporten una corriente algo más elevada que en las demás pistas de conexión normales. Además, como se puede observar en el layout principal (Figura 39), el rutado de las pistas de alimentación está realizado en estrella, de esta forma es posible asegurarse de que no haya una pista que soporte una gran cantidad de corriente durante algún tramo del circuito, ahorrando una posible fuente de problemas al mismo.

Para el rutado de las pistas de conexión, el ancho de las mismas se fijó hasta los 20mils, debido a que la cantidad de corriente que circula por ellas es bastante menor a la que se proporciona a las pistas de alimentación por lo que no hay mayor problema para disminuir el tamaño de éstas, haciendo el rutado más sencillo y economizando en capas para la fabricación final de la placa.

5.1.3.- Pads de los componentes.

El tamaño de los padstacks así como de los huecos de los taladros del circuito también se modificaron para facilitar el soldado del mismo, otro motivo importante para modificarlos era que algunos componentes como por ejemplo los conectores que fijaban el micro a la placa eran demasiado grandes como para entrar por el taladro estándar debido a que las medidas por defecto aplicadas por el programa no tenían el suficiente diámetro, siendo entonces necesaria su modificación.

La configuración de los pads para cada uno de los componentes es la siguiente.

Componente	Padstack	Taladro
Para los encapsulados de los integrados	70	34
Para los conectores del microprocesador	80	40
Para los test point	80	40
Para las resistencias	70	28
Para los condensadores	70	34
Para los conectores	80	40
Tamaño de las vías:	56	28

Como se puede observar el ancho del padstack es como mínimo el doble del ancho del taladro para asegurar que no haya ningún problema a la hora de soldar o desoldar por algún tipo de error.

Es apreciable que los pads tanto para los conectores del microprocesador como para los de alimentación son más elevados que los demás debido a que éstos son físicamente bastante anchos. Las vías en cambio tienen el ancho más pequeño de todos los pads ya que no hace falta que conecten entre sí las rutas de las dos capas que unen mediante cable u otro conector debido a que ya de por sí vienen conectadas interiormente gracias a un baño metálico que pasa por su interior.

5.1.4.- Capas de la PCB.

El diseño de la PCB está realizado en dos capas, TOP y BOTTOM, el uso de ambas capas es necesario para mantener unas dimensiones de la placa dentro de unos límites que permitan su fácil manejo. La capa TOP únicamente es utilizada para rutar pistas, mientras que la capa BOTTOM aparte de ser utilizada para el rutado de las pistas también sirve para imprimir un plano de masa para unificar así todas las conexiones a tierra de todos los circuitos integrados, condensadores y demás componentes del circuito.

5.1.5.- Plano de masa.

El plano de masa, como se ha explicado antes, está situado en la capa BOTTOM de la PCB. El clearance (espacio sin cobre que se deja entre la ruta y el plano que se encuentran en la misma capa) del mismo es de 20mils para evitar así que pueda haber contactos entre el propio plano y alguna ruta a la hora del soldado por su cercanía, además también sirve para facilitar este soldado al dar una mayor maniobrabilidad.

5.2.- Montaje.

En esta fotografía se puede observar tanto la placa recién venida de fábrica como el montaje definitivo de la misma, se puede ver como el nombre de cada uno de los componentes viene xerografiado para tener más facilidad de reconocimiento de cada uno de los mismos a la hora de soldarlo así como para la realización de las pruebas oportunas. Como se puede observar, todos los componentes del circuito son discretos y comunes, así como con tamaños estandarizados, por lo que se pueden encontrar fácilmente en cualquier tienda de componentes.

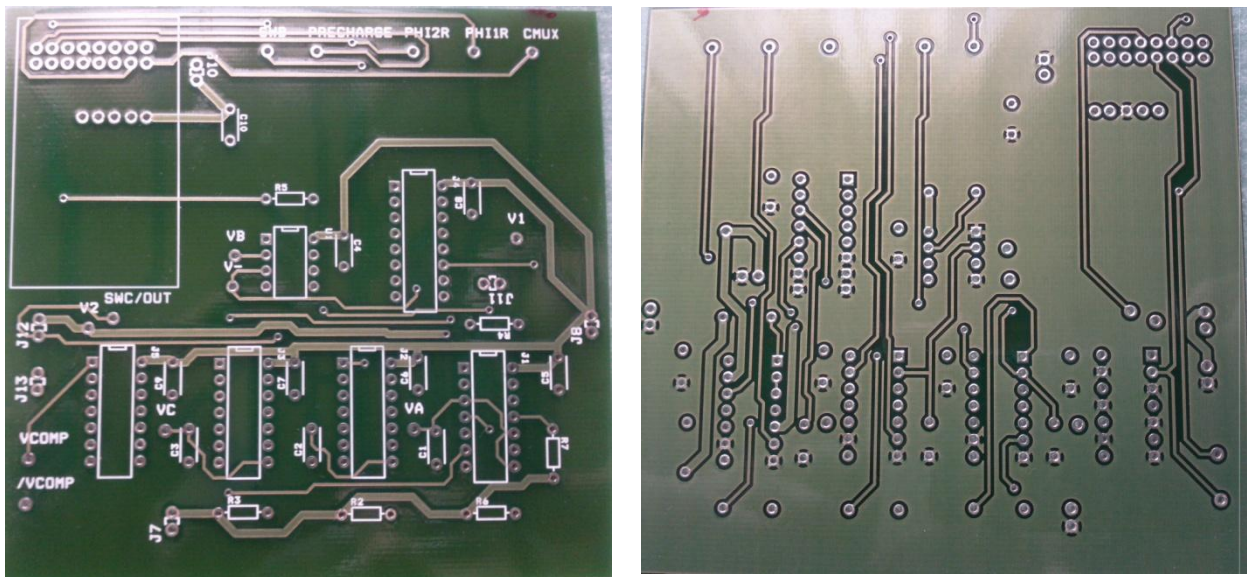


Figura 40: Capas TOP (izquierda) y BOTTOM (derecha) de la PCB fabricada.

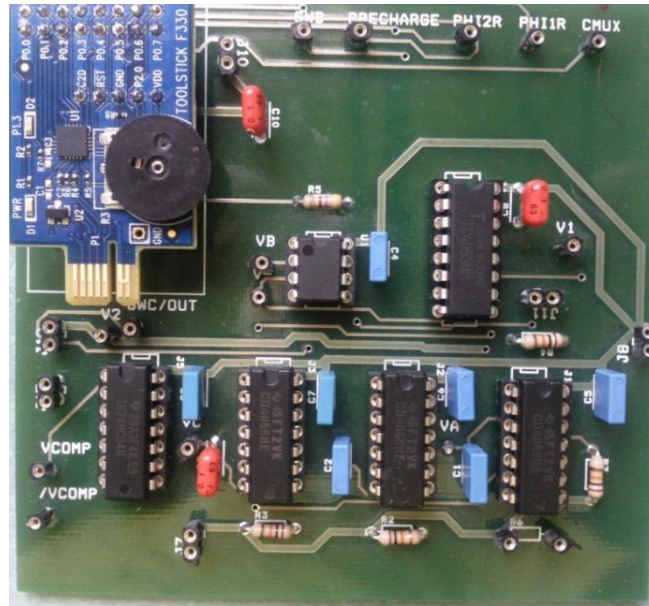


Figura 41: Montaje del circuito soldado y funcional.

Por último mostraremos una foto del circuito funcionando en el puesto de trabajo, con su conexión mediante USB para conectar el micro al ordenador y así programarlo y el osciloscopio usado para las pruebas. Como se puede observar en esta imagen sólo hizo falta el uso de una de las fuentes de alimentación, ya que se podían alimentar tanto los integrados como los condensadores a 5V, la alimentación de la fuente al micro está desconectada ya que mientras éste esté conectado al ordenador se alimenta mediante su conexión serie al mismo mediante USB, por lo que no hace falta alimentarlo externamente.

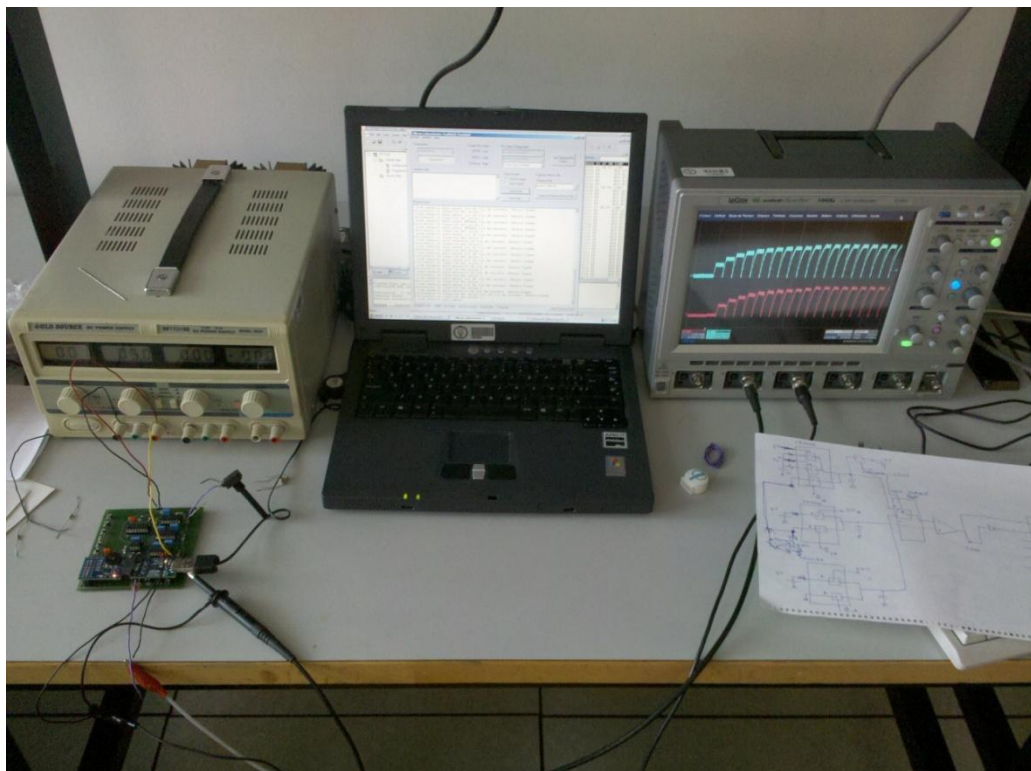


Figura 42: Funcionamiento del circuito en puesto de trabajo.

5.3.- Pruebas.

En este sub-apartado vendrá explicadas todas las pruebas que se realizaron para comprobar que el circuito funcionaba correctamente y para dar el visto bueno a un posible desarrollo futuro de un circuito basado en el demostrador.

5.3.1.- Listado de pruebas a realizar.

-Señales de control-

Prueba N°1: Ph1R y Phi2R.

Prueba N°2: Phi1R y CMUX.

-Configuración de retardo-

Prueba N°3: Carga en los condensadores C_1 y C_2 .

Prueba N°4: Señal de entrada comparada con la señal SWB intermedia.

Prueba N°5: Tensión en C_2 comparado con la salida del comparador.

Prueba N°6: Demostración del retraso de las señales un periodo.

-Configuración de filtro de 1^{er} orden-

Prueba N°7: Evolución de la tensión en los condensadores 2 y 3 con el tiempo.

Prueba N°8: Evolución de los condensadores 1 y 3 en el tiempo.

Prueba N°9: Comparación de la entrada con la evolución de la tensión en el condensador 3.

Prueba N°10: Funcionamiento del integrador con crecimiento rápido y pulso de entrada pequeño en el tiempo.

Prueba N°11: Funcionamiento del integrador con crecimiento medio y pulso de entrada muy pequeño en el tiempo.

Prueba N°12: Funcionamiento del integrador con crecimiento medio y pulso de entrada medio en el tiempo.

Prueba N°13: Funcionamiento del integrador con un crecimiento muy rápido de la señal en el tiempo y una entrada muy pequeña en el tiempo.

Prueba N°14: Funcionamiento del integrador con un crecimiento medio y pulso de entrada muy pequeño en el tiempo.

Prueba N°15: Funcionamiento del integrador tras el paso de un tiempo, cuando se encuentra estabilizado.

5.3.2.- Señales de control.

Prueba N°1:

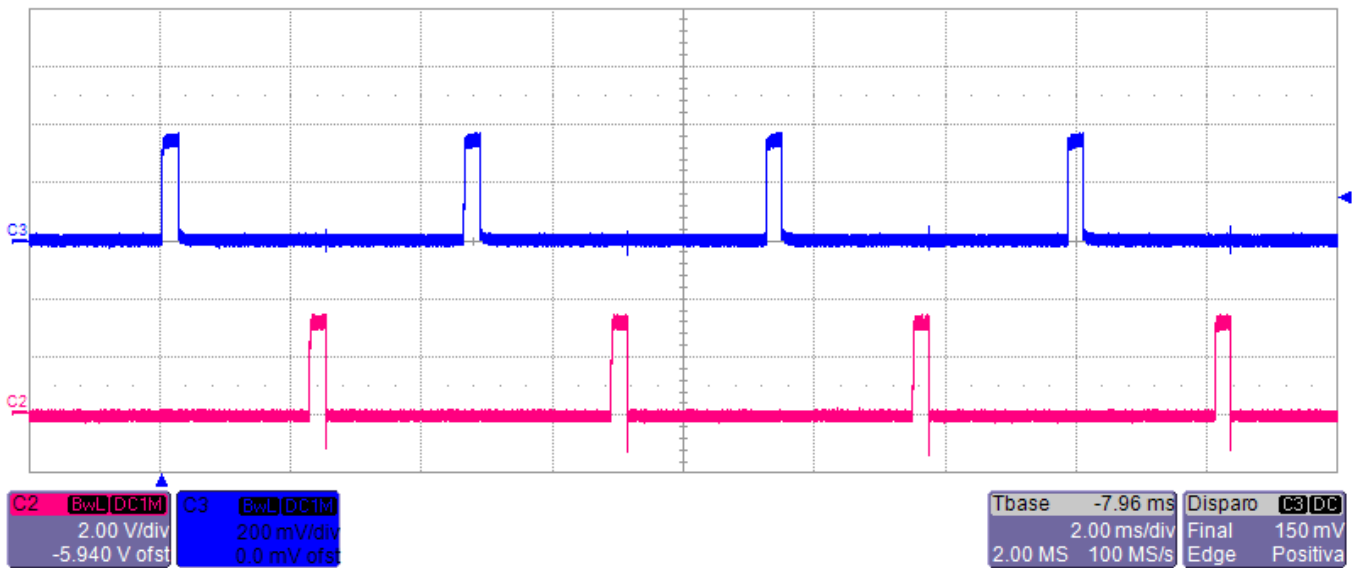


Figura 42.

En esta primera prueba se busca observar el funcionamiento de las señales de control que se encargan de descargar los condensadores, estas serían Phi1R (C3 en la imagen) y Phi2R (C2 en la imagen). Como se puede apreciar, la duración de estas señales es muy estrecha comparado con la duración del periodo, es decir, estas fases son más efímeras que las otras dos, justo como se ha explicado en el apartado de desarrollo del software. Además, se puede intuir de igual forma la duración de las otras dos fases, que es precisamente el hueco existente entre la señal Phi1R en el osciloscopio y la señal Phi2R, en esos espacios en blanco se encuentran las fases 1 y 3.

Para comprobar que las señales son correctas, se puede ver que Phi1R es la que se dispara en un primer momento, tras este pulso, hay un periodo de casi 2ms en los cuales entra en acción la fase 1 del programa, que se encarga de activar la señal SW_c , al acabar, comienza la fase 2 y por lo tanto se dispara la señal Phi2R como se puede apreciar en la captura, el periodo hasta que se vuelve a disparar Phi1R se puede deducir como la fase 3, donde se dispara la señal SW_b .

Prueba N°2:

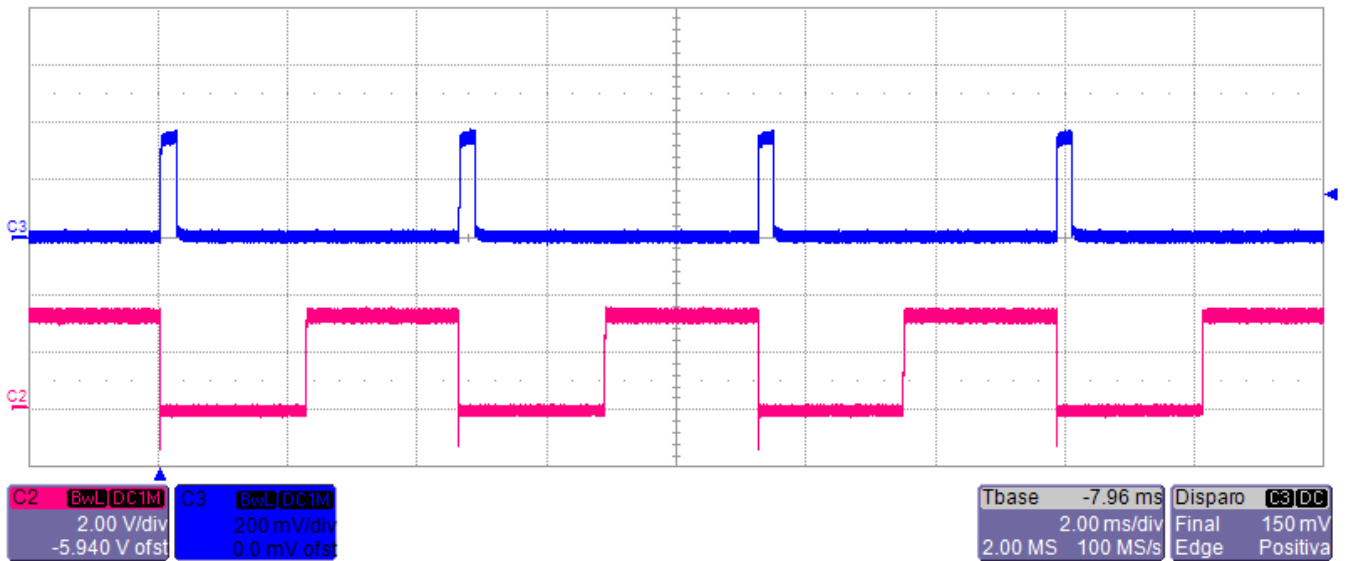


Figura 44.

En esta prueba se puede apreciar cómo se comportan en el tiempo las señales Phi1R y Cmux, precisamente esta captura se puede observar que la señal de reloj de Cmux está bien programada y que el microprocesador la activa de forma correcta.

Para apreciar como efectivamente concuerda con el organigrama explicado anteriormente, es fácil fijarse primero, en que cada uno de los niveles de Cmux ocupa la misma parte del periodo, es decir, que está 2ms a nivel alto, y otros 2ms a nivel bajo, justo como se pretendía en un primer momento a la hora de programar las señales de control. Por otro lado se puede ver como cuando Phi1R se activa, Cmux pasa a nivel bajo, fijándose en el organigrama de la interrupción del timer que controla el A/D se puede observar como justamente está programado para que pase así en la fase 0, luego se puede observar, comparando esta captura con la de la prueba N°1 como las dos señales Phi2R y Cmux pasan a nivel alto en el mismo momento del periodo, es decir, al comienzo de la fase 2.

Con estas capturas queda comprobado como la programación y ejecución de las señales de reloj del circuito se realizan de forma correcta tal y como se supone que deben ejecutarse y que además están perfectamente sincronizadas con los ciclos de trabajo del programa.

5.3.3.- Funcionamiento como célula de retraso.

Prueba N°3:

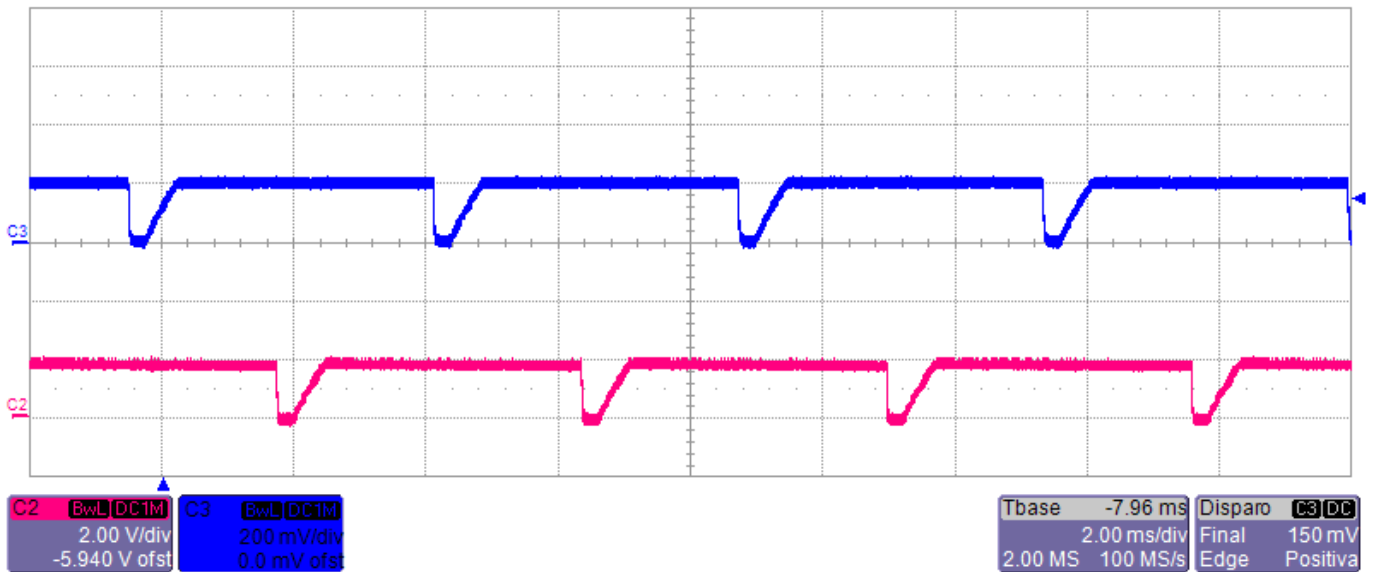


Figura 45.

Para comprobar el funcionamiento del circuito como célula de retraso, la primera prueba a realizar es que los condensadores se carguen a la misma tensión, si lo hacen significa que la mayor parte de la construcción así como la programación del circuito está bien realizada. Quiere decir que primero, funcionan perfectamente las señales de descarga del condensador, segundo, también funcionan las señales de activación de los interruptores de los circuitos de C_2 y C_3 , es decir, las señales SW_b y SW_c , y se activan en el momento en el que deberían para empezar la carga del condensador, tercero, funciona perfectamente la señal de C_{mux} , puesto que el multiplexor elige las señales oportunas correctamente para que el comparador actúe sobre las que debe comparar en cada momento y por último funcionan correctamente las rutinas de interrupción, ya que cuando el comparador manda la señal que indica que ambos condensadores tienen la misma carga, el programa desactiva las señales SW_b o SW_c , según sea oportuno, manteniendo así los condensadores a una tensión exactamente igual que la de su antecesor en el tiempo.

Ateniéndose a la captura del osciloscopio de la prueba N°3, se puede ver como todo lo anteriormente explicado se cumple. Primero que las señales de descarga del condensador actúan bien, el primer condensador (sonda C3) se descarga al inicio del periodo y el segundo (sonda C2) se descarga a la mitad del mismo, siendo el tiempo que los condensadores están con carga nula el momento en el que actúan Φ_{1R} y Φ_{2R} . Luego se puede ver como inmediatamente después empieza la carga de los condensadores, C_1 se carga con la señal de entrada y C_2 en la fase 3, es decir, con la señal SW_b , al final, se puede observar como los dos condensadores acaban exactamente a la misma tensión y además se cargan a la misma velocidad, demostrando así que se ha copiado la carga de C_1 a C_2 .

Prueba N°4:

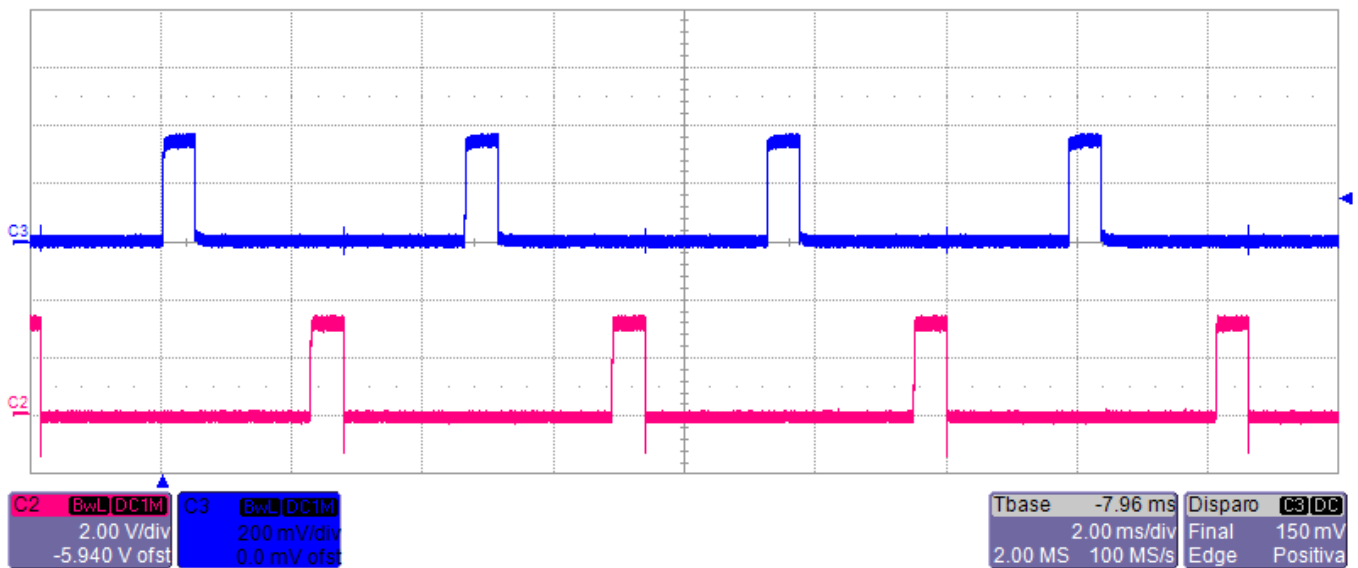


Figura 46.

Esta prueba está realizada exactamente en el mismo punto de funcionamiento que la anterior, pero en este caso, en lugar de las cargas del condensador se ven la señal de entrada al circuito, es decir, la que carga al condensador C_1 , y por el otro lado está la señal SW_b , que es la que se encarga de hacer una carga intermedia en la mitad del periodo de la que se tiene en C_1 . Como se puede observar, el ancho de pulso de ambas es el mismo, esto quiere decir que la copia se ha realizado correctamente, habiéndose memorizado el pulso de entrada, tal y como se esperaba viendo la prueba anterior cuando ambos condensadores tenían la misma carga. Que los condensadores estén cargados a una misma tensión significa que los interruptores que cierran su circuito de carga han estado cerrados exactamente el mismo tiempo mediante las señales de control, en este caso estas señales serían la entrada al circuito simulada por la señal precharge y SW_b . Si se comparan ambas pruebas, se puede observar perfectamente como coinciden el momento ascendente de la carga de los condensadores cuando las señales de esta prueba 4 están a nivel alto, como se ha explicado desde un principio estas cierran los interruptores que hacen que se cargue el condensador correspondiente.

Estas dos pruebas, a falta de observar cómo se comporta el condensador C_3 , son capaces de darnos una idea sobre qué está bien en el circuito y en el programa, que se copien estas dos señales implica que la mayoría de los componentes funcionan bien, también indican que se ha rutado la placa correctamente y que no hay problema con el ancho de las pistas, también nos demuestra que no hay errores de conexión, y que el programa y las salidas el microprocesador funcionan correctamente.

Prueba N°5:

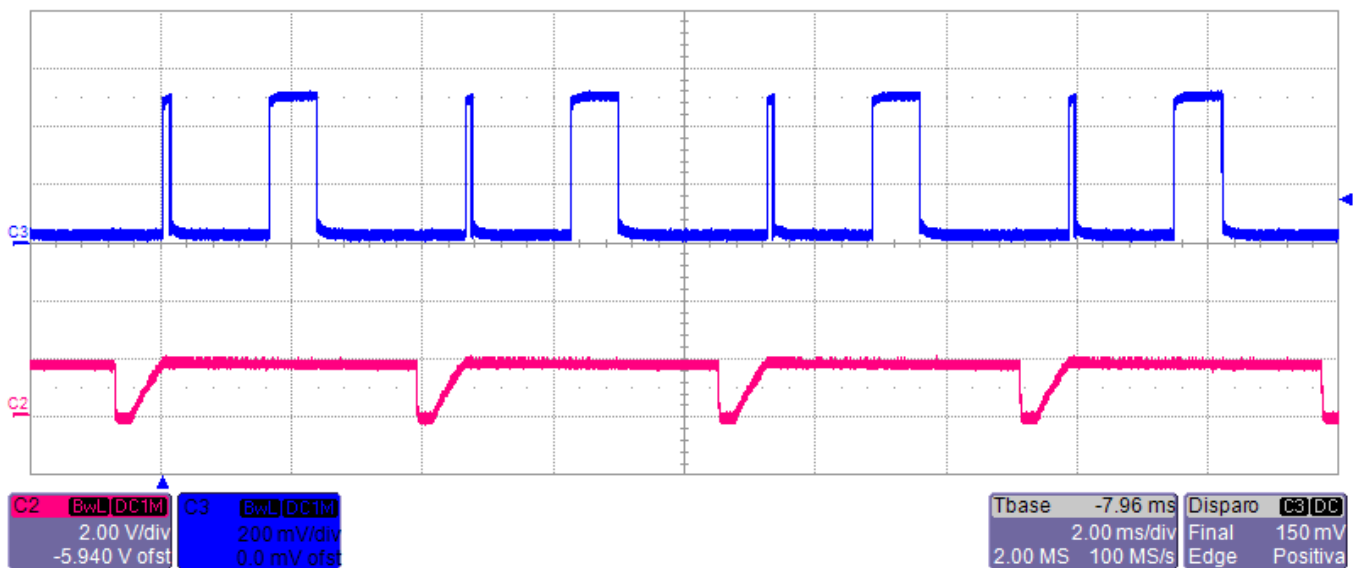


Figura 47.

El objetivo de esta simulación es observar el funcionamiento del comparador cuando uno de los condensadores llega a la carga a la que está el condensador al que desea copiar su tensión. Para esta simulación se ha usado la carga del condensador C_2 ya que el pulso que debe dar el comparador cuando éste copia a la carga de C_1 es positivo y por lo tanto es más fácilmente visible, para el caso del condensador C_3 la permutación de tensiones es justo al contrario, ya que la tensión de éste está conectada a la patilla negativa del comparador así como C_1 (coordinadas por el multiplexor), pero a diferencia del condensador C_1 , C_3 tiene que superar en tensión a C_2 , y por lo tanto en la cuanto supere el comparador pondrá su salida a nivel bajo.

Pero en el que caso que atañe a esta captura del osciloscopio no es así, en este caso se ve como en el momento en el que el condensador C_2 llega a la misma tensión que el condensador C_1 , el comparador da un pulso positivo, en este momento este pulso llega a la patilla de interrupción externa 1 del microprocesador, el cual mediante software abre el interruptor que carga al condensador C_2 , y como se puede apreciar perfectamente en la captura de pantalla, en cuanto se da el pulso el condensador deja de cargarse y mantiene una tensión constante, específicamente la tensión a la que el condensador C_1 ha sido cargado anteriormente.

Prueba N°6:

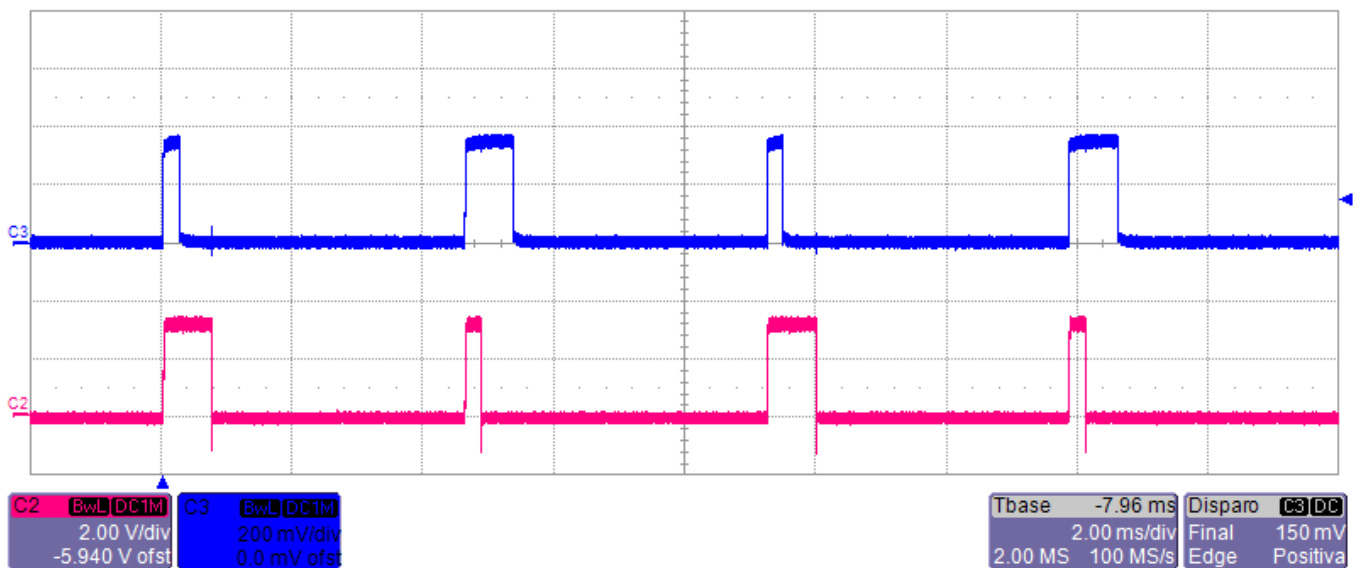


Figura 48.

Esta prueba es la última realizada para comprobar que el funcionamiento del circuito como la célula de retraso es correcto, para ello entra por primera vez en juego el funcionamiento del condensador C_3 ya que se carga con la señal retrasada un periodo.

Para que pudiera ser fácilmente visible esta prueba, se realizó una pequeña modificación en el programa como se explicó en el capítulo de software [4.1.2B], en la que en cada uno de los periodos del circuito se permutara su pulso de entrada alternamente. El motivo de realizar esta modificación es que si mantuviéramos constantemente el mismo pulso de entrada al circuito sería imposible apreciar si se ha realizado un retraso o no. Lo que ocurre ahora es que el programa sigue pidiendo al usuario un ancho de pulso inicialmente, pero en el siguiente periodo lo cambia por otro ancho de pulso distinto haciendo esto alternamente según los periodos pasan, de tal forma que sea mucho más fácil apreciar los retrasos. Simplemente lo que hace la modificación del programa es intercambiar anchos de pulso activando una variable cuando se llega a la última fase de cada periodo, en ese momento la señal precharge que es la que controla el ancho de pulso de entrada es modificada con un valor prefijado, para luego volver a adquirir el valor del pulso de entrada introducido por teclado en el siguiente periodo.

En la captura del osciloscopio realizada en esta prueba, la sonda C3 es la que muestra la señal que carga el condensador C_1 (precharge), mientras que la sonda C2 muestra la que carga al condensador C_3 (SW_c). Como se puede observar, cada pulso de control del condensador C_1 está repetido en el condensador C_3 pero un periodo después, es decir, 4ms después. Con lo cual podemos decir sin ninguna duda que la célula de retraso funciona perfectamente.

5.3.4.- Funcionamiento como filtro de primer orden integrador.

Prueba N°7:

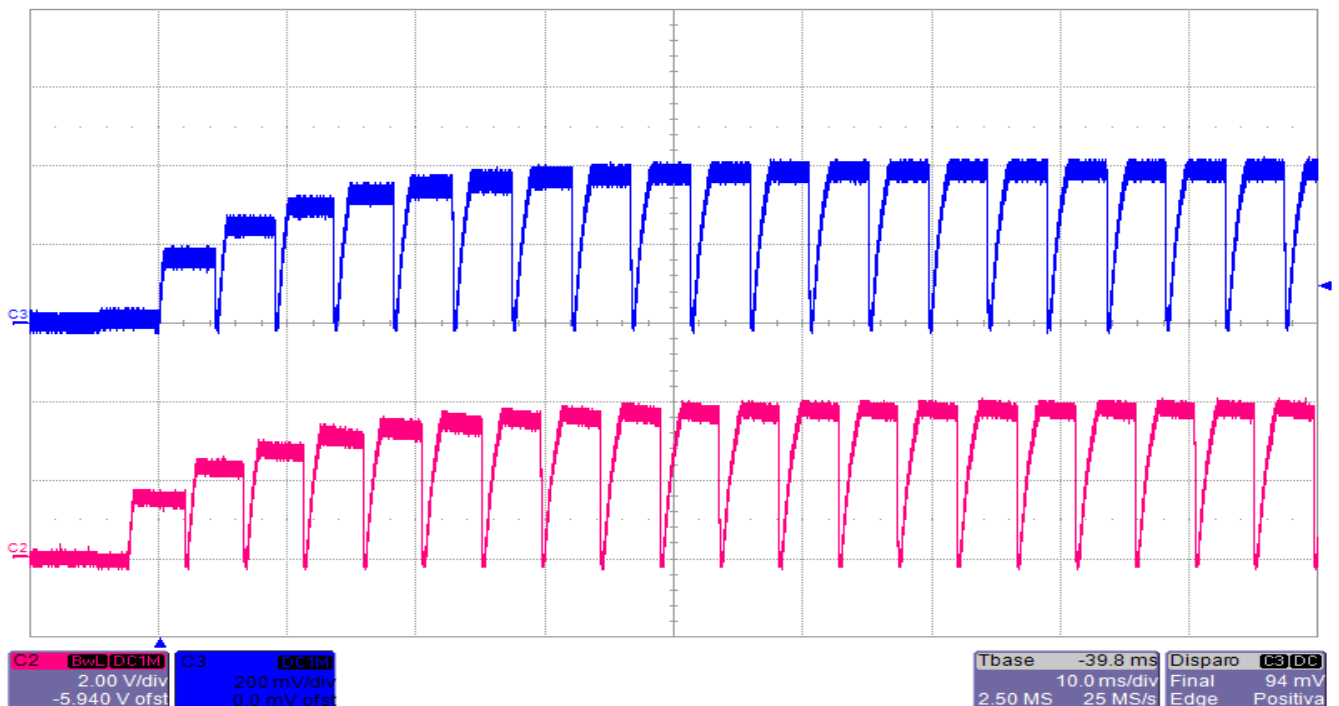


Figura 49.

La prueba N°7 se encarga de demostrar la evolución de la tensión en los condensadores del circuito cuando actúa como filtro de primer orden. Por ejemplo como se muestra en esta captura los condensadores C_1 y C_2 al introducir como entrada un tren de pulsos al demostrador, actuarán incrementando su tensión de carga exponencialmente en el tiempo como haría un sistema de primer orden. Se puede apreciar como la carga de los condensadores es incrementada cada uno de los periodos de forma que al llegar al ancho de pulso máximo permitido esta carga se mantiene constante. Con esta captura se puede observar como el crecimiento exponencial del ancho de pulso a la salida del circuito corresponde con un crecimiento exponencial de la carga del condensador en cada uno de los periodos.

En este caso la señal correspondiente al condensador C_1 se ve con la sonda C2 mientras que la señal correspondiente al condensador C_2 con la C3. Esta puntualización es importante a la hora de observar como el condensador C_2 empieza a funcionar justamente cuando el condensador C_1 lleva medio periodo cargándose al igual que hacía como cuando funcionaba como célula de retraso, ya que durante su funcionamiento como filtro las señales de control y su secuencia de funcionamiento no se ve modificada.

Prueba N°8:

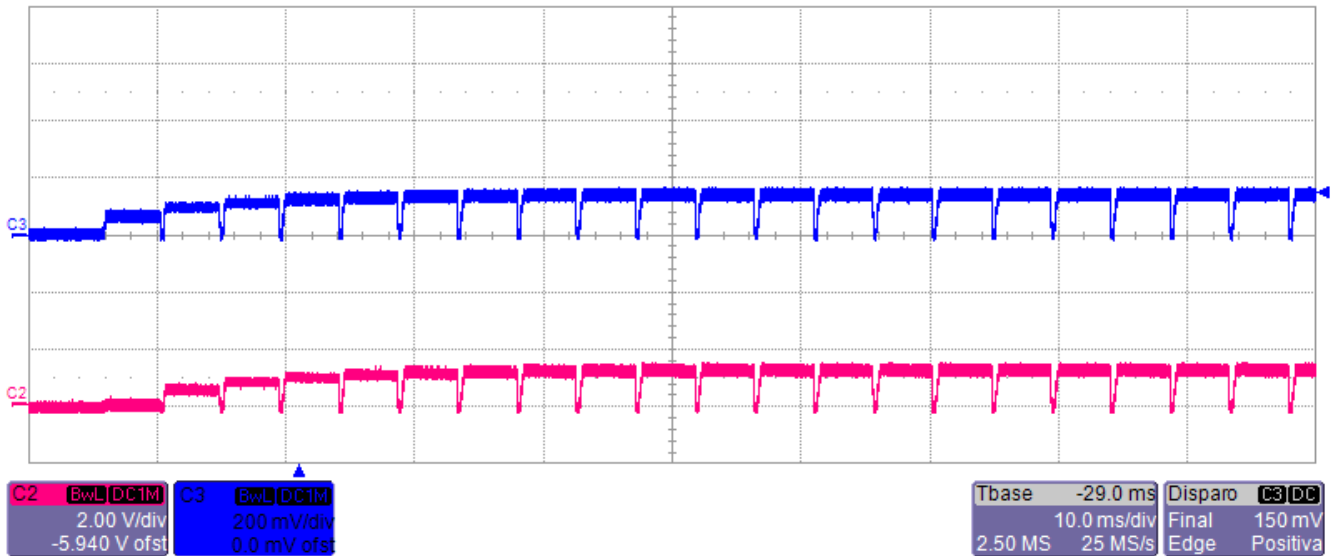


Figura 50.

Esta prueba simplemente sirve para comprobar que se cumple la premisa de retraso de 1 periodo en el funcionamiento del condensador C_3 con respecto al condensador C_1 , incluso en el demostrador funcionando como filtro de primer orden. Siendo la sonda C3 la que muestra la secuencia de carga en el condensador C_1 y la sonda C2 la de la secuencia de carga en el condensador C_3 , se puede apreciar perfectamente que el crecimiento en la señal en este último condensador comienza justamente después de que el primero haya tenido una primera carga, probando que el funcionamiento de la célula de retraso funcionando como bloque dentro del filtro de primer orden es correcto.

Prueba N°9:

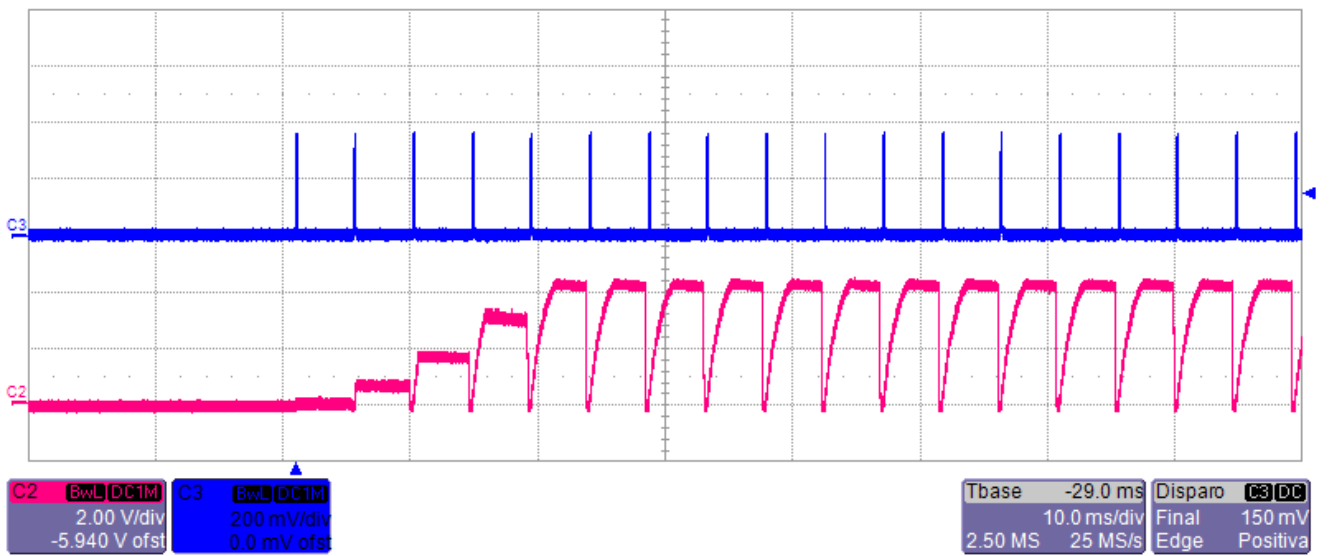


Figura 51.

El objetivo de esta simulación es principalmente observar la evolución de la carga en el condensador C_3 mientras en la entrada del circuito hay un tren de pulsos de ancho continuo. Como se puede ver, la carga en el condensador C_3 va evolucionando hacia valores superiores de tensión debido a la realimentación que tiene el demostrador que le proporciona un funcionamiento de integrador, el pulso de entrada se complementa con el de salida realimentado haciendo que en cada periodo el circuito de carga RC esté un mayor tiempo funcionando, además, durante el tiempo que tanto el interruptor de entrada como el de realimentación están cerrados, la constante de tiempo del circuito se modifica haciéndolo más rápido (el paralelo de las dos resistencias da una resistencia menor vista por el condensador, y así una menor constante de tiempo) y por lo tanto alcanzando durante ese tiempo una carga mayor a la que anteriormente tenía.

Prueba Nº 10:

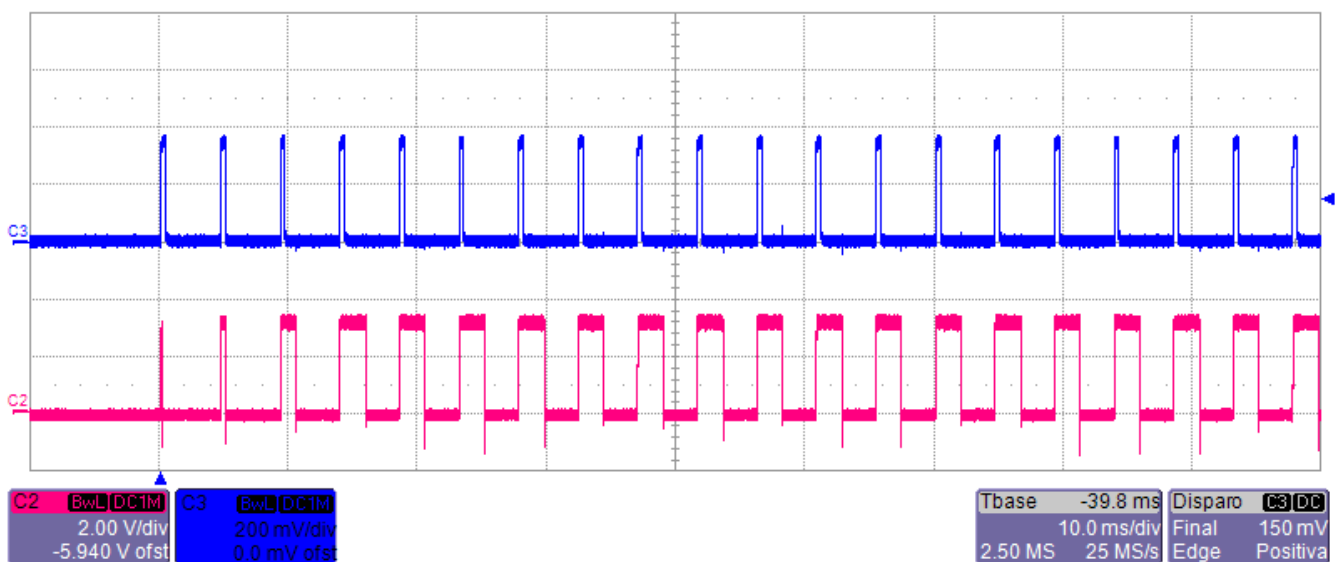


Figura 52.

A partir de esta prueba número 10, las señales que se ven son siempre la entrada (señal de arriba) y salida del circuito (señal de abajo), haciendo varias pruebas modificando tanto el ancho de pulso de la entrada al circuito como el valor de la variable p que interviene en la posición del polo para observar su respuesta.

En esta primera simulación, se quiere ver el funcionamiento del demostrador como filtro de primer orden integrador introduciendo una señal de entrada estrecha en el tiempo y buscando un crecimiento rápido del circuito. Como se puede ver en la imagen, la estabilización del circuito es extraordinariamente rápida, prácticamente en dos periodos ha llegado a la señal con ancho definitivo, esto es debido principalmente a la posición en la que se ha colocado el polo la cual es cercana al eje y no nos deja prácticamente ver cómo funciona el filtro, por lo tanto, si se quiere comprobar mejor el funcionamiento del circuito como integrador de una manera más exhaustiva se podrían hacer dos modificaciones básicas. La primera es la modificación del pulso de entrada haciéndolo más pequeño y por lo tanto cargando el condensador a una menor tensión en

cada periodo, y por otro lado la modificación de la variable p de tal forma que el crecimiento exponencial sea lo más progresivo posible.

En las siguientes pruebas del circuito, lo que se hará será modificar estos dos parámetros viendo así la capacidad de reacción que tiene el circuito y cómo evoluciona su señal de salida al modificarlos.

Prueba N°11:

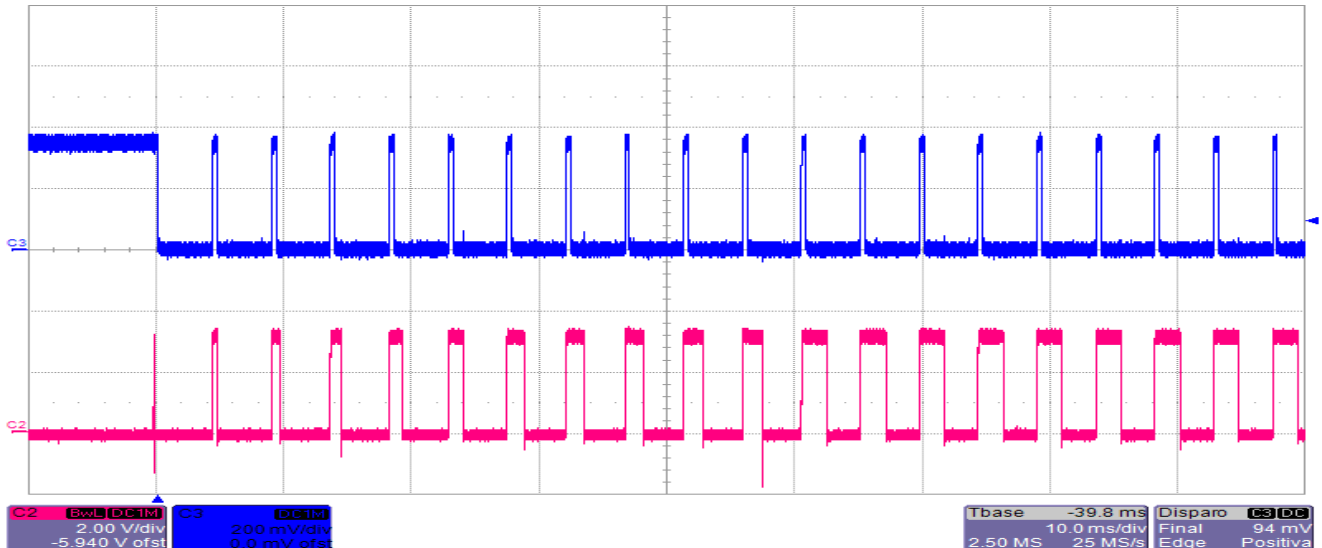


Figura 53.

En esta simulación, lo único que se ha modificado en el circuito es el valor del parámetro p , el pulso de entrada sigue teniendo el mismo ancho de pulso que en la simulación anterior. Pero como se puede apreciar, la simple modificación del valor de p puede cambiar completamente el resultado obtenido a la salida del circuito, el crecimiento es mucho más suave que en la anterior simulación, creciendo durante unos diez periodos de forma exponencial hasta que llega al máximo, haciendo así mucho más visible el comportamiento como filtro integrador.

Prueba N°12:

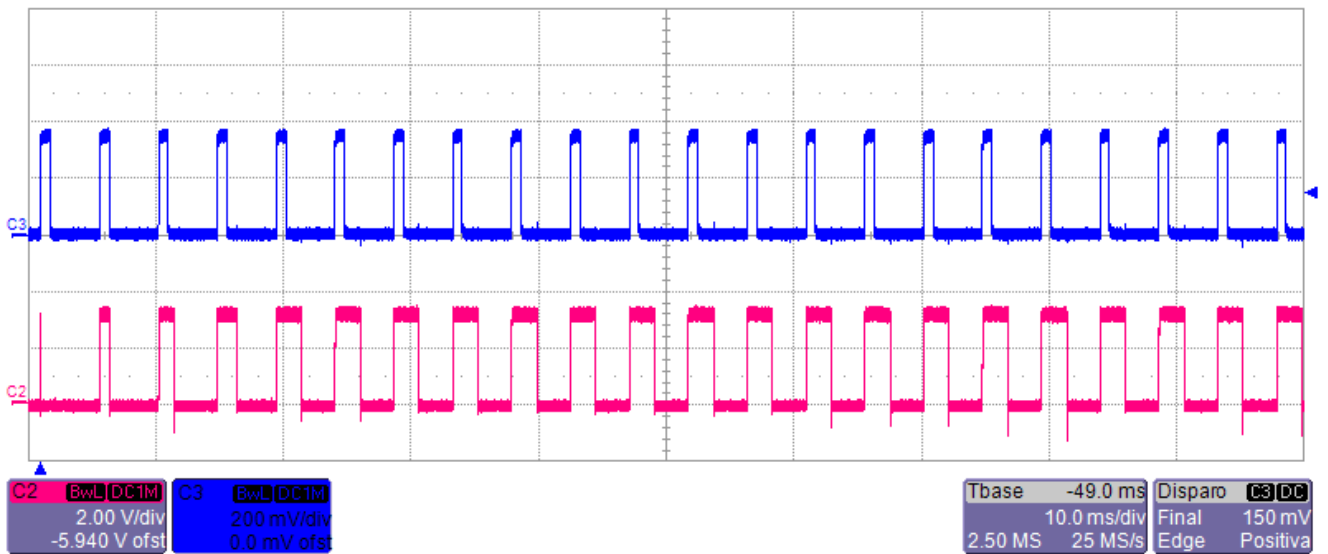


Figura 54.

Manteniendo el mismo valor de la constante p de la prueba anterior, en esta prueba lo que difiere es el ancho de pulso de entrada, como se ha explicado en la prueba 10 la única forma de modificar el comportamiento del circuito es modificando su señal de entrada o en cambio modificando el valor del parámetro p ya que g se mantiene constante. En este caso la modificación se realiza a la entrada introduciendo al circuito un pulso más ancho en el tiempo que anteriormente. Como se puede observar por lo tanto, la estabilización y crecimiento de la señal de salida es mucho más rápida que en la prueba anterior.

Prueba N°13:

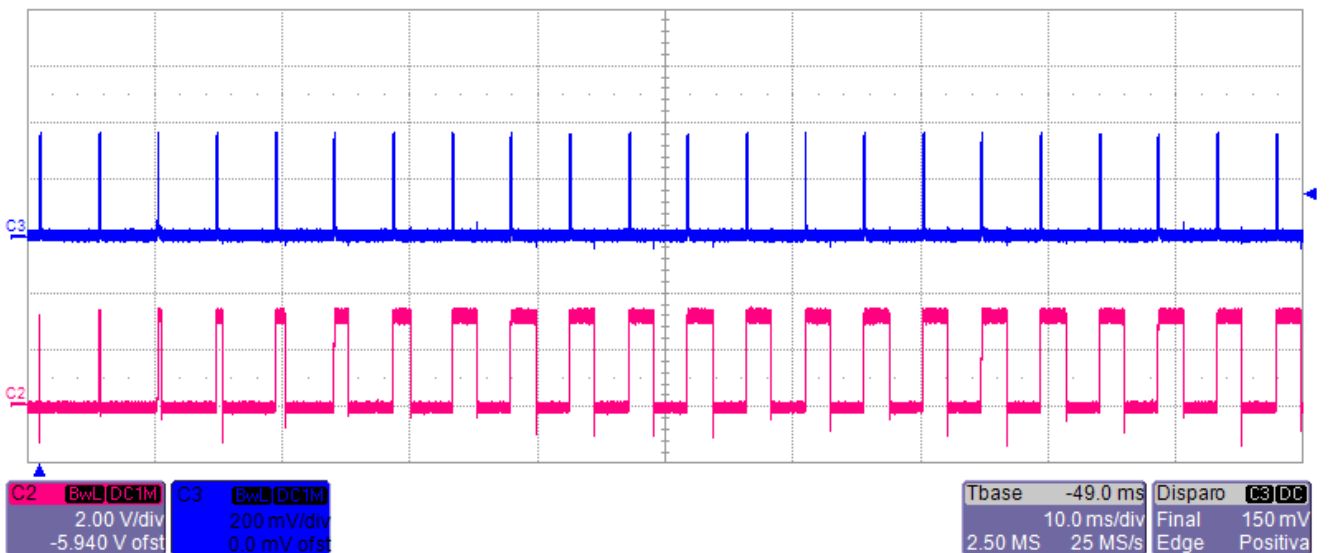


Figura 55.

En esta prueba, se modificaron los dos parámetros del circuito para demostrar cómo es posible obtener una respuesta similar que en casos anteriores en condiciones completamente distintas, en este caso obtener un resultado parecido al obtenido en la prueba N°11, como se puede observar la entrada es mucho más pequeña que en la prueba mencionada, por lo tanto, al circuito le costaría mucho más llegar a una estabilización tan rápida como en la N°11 manteniendo un crecimiento más o menos medio, de esta forma, para intentar igualar a esa mayor entrada el único medio que se puede utilizar es modificar p para hacer al circuito mucho más rápido, que es lo que ocurre con esta prueba, como se puede ver modificando p y por lo tanto cambiando la posición del polo, el filtro puede llegar a ser igual de rápido en condiciones completamente distintas de entrada al anterior.

Prueba N°14:

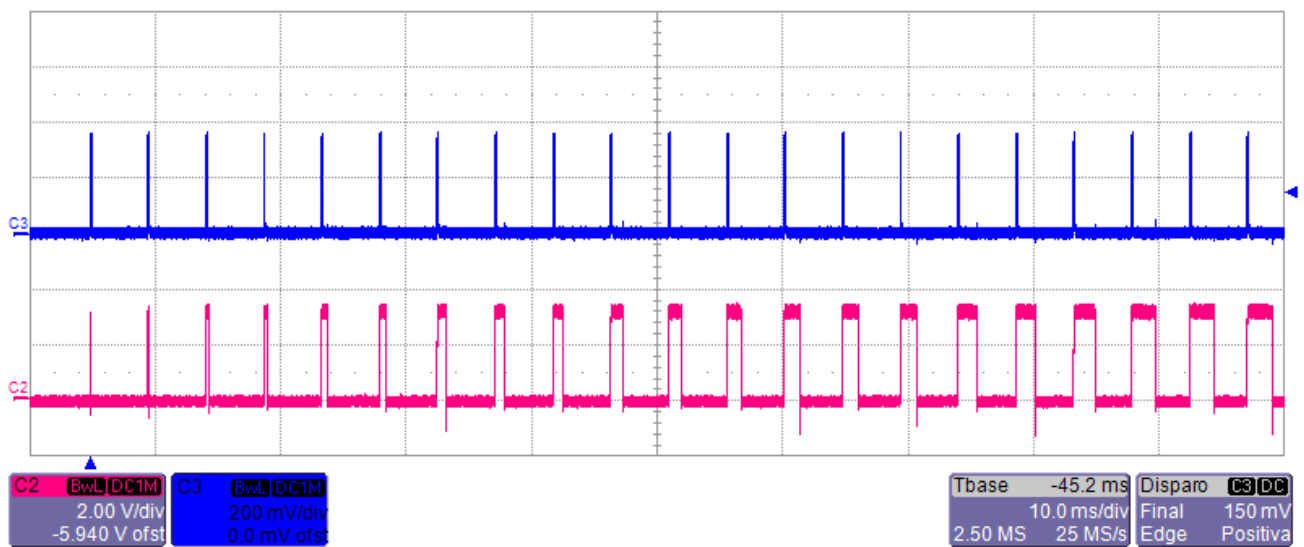


Figura 56.

La realización de esta prueba se hizo para intentar encontrar unos parámetros justos para que se pudiera observar perfectamente el crecimiento exponencial de la salida y el funcionamiento del circuito como integrador, para ello se introdujo a la entrada el mínimo pulso que por software tiene posibilidad de crear para que así empezara a funcionar desde el nivel más bajo, buscando luego un comportamiento de p que hiciera que el crecimiento fuera perfecto para su visualización. Como se puede observar en la imagen por cada periodo del circuito el ancho de pulso crece un poco exponencialmente de forma que al final alcanza un ancho de pulso estabilizado, pero antes de llegar a éste la salida pasa por casi veinte periodos de crecimiento, suavizando mucho más el funcionamiento del circuito y la acomodación de la señal.

Prueba N°15:

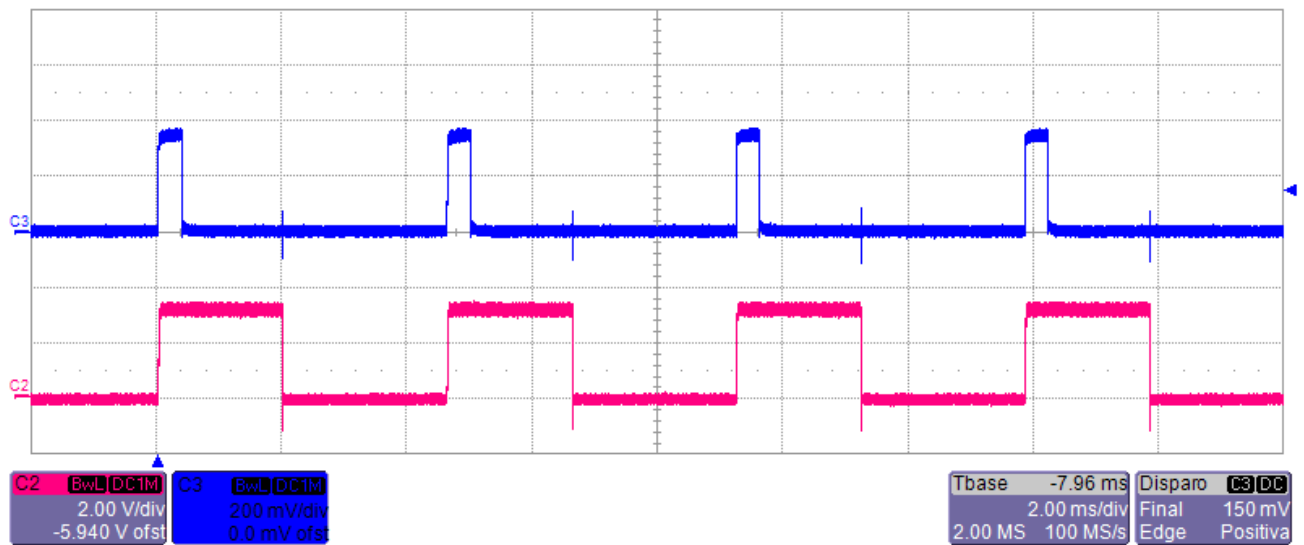


Figura 57.

Esta última prueba se ha realizado simplemente para observar el funcionamiento del filtro de primer orden una vez que se ha estabilizado el circuito, como se puede observar en la imagen, la salida del circuito (señal C2), se mantiene constante, con un ancho de pulso mayor que el de la entrada, esto es debido a que mediante la realimentación, el circuito llega a un punto en el que su salida no puede crecer más, la causa de esto son tres principales motivos: El primero como se explicó en el capítulo 3 es que la señal C_{mux} limita la carga de los condensadores, el segundo es que las señales Φ_{1R} y Φ_{2R} borran periódicamente la carga de los mismos no dejando que se puedan cargar ilimitadamente, y el tercero, aunque los condensadores tuvieran mucho tiempo para cargarse sólo llegarían a la tensión a la que están alimentados y no podrían tener una mayor carga y como consecuencia un ancho de pulso mayor.



6.- Conclusión.



Siendo el objetivo del proyecto que la creación de un circuito demostrador que realizara una serie de tareas para ver si son posibles de alcanzar y así poder en un futuro crear un chip, las conclusiones sobre éste son muy positivas en cuanto a que la idea y el comportamiento del circuito es tal como se esperaba, es decir, el circuito funciona tal y como se suponía que lo haría una vez fue ideado.

Lógicamente a la hora de analizar los resultados del circuito hay que tener en cuenta que las frecuencias de funcionamiento de este circuito son muy pequeñas en comparación con las frecuencias a las que funcionaría realmente, como se ha explicado las frecuencias de funcionamiento del demostrador son de 1KHz aproximadamente, y en aplicaciones reales, ésta se elevaría hasta los 25MHz, sin lugar a dudas la diferencia es abismal, pero es interesante percatarse que para la creación de las señales de reloj se utiliza un microprocesador y éste tampoco puede realizar una frecuencia de reloj con su interrupción mucho mayor a la utilizada, mientras que en un circuito real la frecuencia de reloj vendría dada por un verdadero circuito de reloj funcionando a altas velocidades, sin embargo los resultados pueden ser considerados como extrapolables, ya que lo importante, que es el principio de funcionamiento sería exactamente igual tanto para bajas frecuencias como para altas frecuencias. Así pues el resultado de funcionamiento a bajas frecuencias es realmente bueno mediante el uso de un microprocesador.

En cuanto a los resultados de la fabricación del layout y el soldado de los componentes, éstos fueron muy positivos teniendo en cuenta que la creación de la PCB se tuvo que hacer sin poder simular el circuito, simplemente con conectores genéricos debido a la falta de librerías en el simulador, por lo tanto siempre hubo cierto grado de incertidumbre hasta que se pudo probar el circuito definitivamente y ver que funcionaba bien.

En definitiva, y como reflexión final, nos quedamos principalmente con que el objetivo con el que se empezó el proyecto y todo lo que teníamos en mente en cuanto al comportamiento que éste debía tener se cumplió con creces, el circuito funcionó exactamente como debía con unos resultados que incluso mejoraban las simulaciones realizadas por OrCAD, y por lo tanto pudiendo dar cierta vía libre al diseño y realización del circuito en tecnologías de fabricación por chip ya sean de 0.35 micras o de 65 nanos con la certeza de que puede funcionar e implementarse en un circuito.



7.- Bibliografía.



Oppenheim, Alan V., Willsky, Alan S. “Señales y Sistemas Segunda Edición” Editorial Pearson 1998.

Hernández, L. “Discrete Time Sigma Delta Modulators based on Time Encoding Integration” 2009 Paper.

Hernández, L. “Time Encoded Signal Processing applied to Sigma Delta Modulation” 2009 Paper.

Hernández, L., Fernández, J., Prefasi, E., Patón S. “A Time Encoded Decimation Filter for Noise Shaped Power DACs” Paper



8.- Presupuesto

Cód. de integrado	Denominación	Descripción	Cantidad	Precio	Importe total
4053	C. Integrado Mod.4053	Triple Multiplexor/Demultiplexor analógico.	3	0,2000 €	0,6000 €
4066	C. Integrado Mod.4066	Cuádruple interruptor analógico/Cuádruple multiplexor.	7	0,1500 €	1,0500 €
74HC04	C. Integrado Mod.74HC-04	Circuito integrado con 6 inversores digitales.	3	0,1207 €	0,3621 €
LM311TC	C. Integrado Mod.LM 311 TC	Comparador de voltaje de alta velocidad.	2	0,1200 €	0,2400 €
LM324	C. Integrado Mod.LM 324	Amplificador operacional de alta ganancia.	2	0,1100 €	0,2200 €
PCB	Placa PCB	Placa PCB a doble capa con plano de masa y nombres.	1	300,0000 €	300,0000 €
Total					302,4721 €



9.- Anexos.



9.1.- Listado de programa.

9.1.1.- Código fuente del programa original.

```
//-----
// PFC Sergio López Batres.c
//-----
// Copyright (C) 2008 UC3M.
//
// AUTH: LH, SL
// DATE: 3 FEB 2010
//
// Este programa contiene las cabeceras del programa principal,
// la configuracion y las rutinas de interrupcion del 8051F330
//
// Target: C8051F330/1/2/3/4/5
//
// Tool chain: KEIL Eval 'c'
//

//-----
// Includes
//-----
#include "C8051F330.h"
#include <stdio.h>

unsigned int cont_phase=0,cont1=0,cont2=0,cont3=0,cont4=0,ancho=0; //declare aqui
las variables enteras de su programa (sin signo)
unsigned char flag='0',flag1='0';flag2='0',flagprueba='0'; lp=6; //declare aqui las
variables byte o char de su programa (sin signo)

sbit LED = P1^3;
sbit precharge = P1^0;
sbit phi2r = P1^1;
sbit swb = P1^2;
sbit phi1r = P0^6;
sbit cmux = P0^7;
sbit swc = P1^4;
sbit PIN_INT0 = P0^2;
sbit PIN_INT1 = P0^3;

void Init_Device(void);

//-----
// MAIN Routine
// NO TOQUE EL CODIGO DE AQUI HASTA LA SIGUIENTE MARCA
//-----
main (void) {
Init_Device();
TI0=1;
```



PCA0MD &= ~0x40;

// INSERTE EL CODIGO DE SU PROGRAMA PRINCIPAL AQUI. RECUERDE QUE TIENE QUE SER UN BUCLE SIN FIN

```
while (1) {

    if(flag=='0')
    {
        printf("\nProyecto de fin de carrera: Sergio Lopez");
        precharge=0;
        flag='1';
    }
    printf("\nintroduzca la longitud del pulso ");
    lp=getchar()-'0';

}
```

// FIN DEL PROGRAMA PRINCIPAL

// INSERTE AQUI LA RUTINA DE INTERRUPCION DEL ADC. SE DISPARA AUTOMATICAMENTE

```
void ADC0_ISR (void) interrupt 10
{
```

```
    if(cont_phase==0)
    {
```

flag1='0'; //Usamos tanto flag 1 como flag 2 para que la activación de los interruptores sea lo más rápida posible.

```
        flag2='0';
        if(cont1<2)
```

{ // El contador lo se pondrá a 8 máximo, para que la frecuencia de funcionamiento del circuito sea 1khz, ya que 9 interrupciones timer AC equivalen aproximadamente a 1ms.

```
            phi1r=1; //Activamos el pin para borrar el
            condensador
            cont1++;
            cmux=0;
        }
```

```
    else
```

```
    {
        phi1r=0; //siguiente fase incrementando el
```

contador de fase.

```
        cont_phase++;
        cont1=0;
    }
```

```
}
```




```

if(cont_phase==1)
{
    if(cont2<14)
    {
        if(flag1=='0')//Haciendo esto conseguimos que la
activación del interruptor sea lo más rápida posible para que interfiera lo mínimo
en la copia de señales
        {
            swc=1; //Activamos el tercer circuito de
interruptores para copiar la señal anterior
            flag1='1';
        }
        if(cont2<1p)
        {
            precharge=1; //En la fase 1 es cuando se ha de
cargar el condensador A a la tensión que queramos, ya que durante la fase 0, éste
se borra, lo cargamos gracias a la señal precharge.
        }
        else
        {
            precharge=0;
        }
        EA=1;
        cont2++;
    }
    else
    {
        cont_phase++;
        cont2=0;
    }
}
if(cont_phase==2)
{
    if(cont3<2)
    {
        phi2r=1;
        cont3++;
        cmux=1;
    }
    else
    {
        phi2r=0;
        cont_phase++;
        cont3=0;
    }
}
if(cont_phase==3)
{
    if(cont4<14)
    {

```



```

        if(flag2=='0')
        {
            swb=1;
            flag2='1';
        }
        EA=1;
        cont4++;
    }
    else
    {
        cont4=0;
        cont_phase=0;
    }
}

AD0INT=0;
}

// INSERTE AQUI LA RUTINA DE INTERRUPCION DEL TIMER 0. SE
// DISPARA AUTOMATICAMENTE 31 VECES/SEGUNDO
void Timer0_ISR (void) interrupt 1
{

}

// INSERTE AQUI LA RUTINA DE INTERRUPCION DE LA INTERRUPCION
// 0. SE ACTIVA POR FLANCO DE BAJADA
void INT0_ISR (void) interrupt 0
{
    swc=0;
    EA=0;
}

// INSERTE AQUI LA RUTINA DE INTERRUPCION DE LA INTERRUPCION
// 1. SE ACTIVA POR FLANCO DE BAJADA
void INT1_ISR (void) interrupt 2
{
    EA=0;
    swb=0;
}

```



9.1.2.- Código fuente del programa para probar el circuito en laboratorio.

```
//-----
// PFC Sergio López Batres.c
//-----
// Copyright (C) 2008 UC3M.
//
// AUTH: LH, SL
// DATE: 3 FEB 2010
//
// Este programa contiene las cabeceras del programa principal,
// la configuracion y las rutinas de interrupcion del 8051F330
//
// Target: C8051F330/1/2/3/4/5
//
// Tool chain: KEIL Eval 'c'
//

//-----
// Includes
//-----
#include "C8051F330.h"
#include <stdio.h>

unsigned int cont_phase=0,cont1=0,cont2=0,cont3=0,cont4=0,ancho=0; //declare aqui
las variables enteras de su programa (sin signo)
unsigned char flag='0',flag1='0';flag2='0',flagprueba='0'; lp=6; //declare aqui las
variables byte o char de su programa (sin signo)

sbit LED = P1^3;
sbit precharge = P1^0;
sbit phi2r = P1^1;
sbit swb = P1^2;
sbit phi1r = P0^6;
sbit cmux = P0^7;
sbit swc = P1^4;
sbit PIN_INT0 = P0^2;
sbit PIN_INT1 = P0^3;

void Init_Device(void);

//-----
// MAIN Routine
// NO TOQUE EL CODIGO DE AQUI HASTA LA SIGUIENTE MARCA
//-----
main (void) {
Init_Device();
EA=0;
TI0=1;
```



```
PCA0MD &= ~0x40;
```

// INSERTE EL CODIGO DE SU PROGRAMA PRINCIPAL AQUI. RECUERDE QUE TIENE QUE SER UN BUCLE SIN FIN

```
while (1) {

    if(flag=='0')
    {
        precharge=0;
        printf("\nProyecto de fin de carrera: Sergio Lopez");
        printf("\nintroduzca la longitud del pulso ");
        lp=getchar()-'0';
        flag='1';
        EA=1;
        printf("\n Pulse espacio para parar");
        if(getchar()==' ') EA=0;
    }

}
```

// FIN DEL PROGRAMA PRINCIPAL

// INSERTE AQUI LA RUTINA DE INTERRUPCION DEL ADC. SE DISPARA AUTOMATICAMENTE

```
void ADC0_ISR (void) interrupt 10
{
```

```
    if(cont_phase==0)
    {
```

flag1='0'; //Usamos tanto flag 1 como flag 2 para que la activación de los interruptores sea lo más rápida posible.

```
        flag2='0';
```

if(cont1<2){ // El contador lo ponemos a 8 máximo, para que la frecuencia de funcionamiento del circuito sea 1khz, ya que 9 interrupciones del timer AC equivalen aproximadamente a 1ms.

philr=1; //Activamos el pin para borrar el condensador

```
        cont1++;
        cmux=0;
    }
```

```
    else
```

{ philr=0; //siguiente fase incrementando el contador de fase.

```
        cont_phase++;
        cont1=0;
```

```

    }
}
if(cont_phase==1)
{
    if(cont2<16)
    {
        if(flag1=='0')//Haciendo esto conseguimos que la
activación del interruptor sea lo más rápida posible para que interfiera lo mínimo
en la copia de señales
        {
            swc=1; //Activamos el tercer circuito de
interruptores para copiar la señal anterior
            flag1='1';
        }
        if(cont2<lp)
        {
            precharge=1; //En la fase 1 es cuando se
ha de cargar el condensador A a la tensión que queramos, ya que durante la fase 0,
éste se borra, lo cargamos gracias a la señal precharge.
        }
        else
        {
            precharge=0;
        }

        cont2++;
    }
    else
    {
        cont_phase++;
        cont2=0;
    }
}
if(cont_phase==2)
{
    if(cont3<2)
    {
        phi2r=1;
        cont3++;
        cmux=1;
    }
    else
    {
        phi2r=0;
        cont_phase++;
        cont3=0;
    }
}
if(cont_phase==3)
{

```



```

        if(cont4<16)
        {
            if(flag2=='0')
            {
                swb=1;
                flag2='1';
            }
            cont4++;
        }
        else
        {
            cont4=0;
            cont_phase=0;
        }
    }
    AD0INT=0;
}

// INSERTE AQUI LA RUTINA DE INTERRUPCION DEL TIMER 0. SE
// DISPARA AUTOMATICAMENTE 31 VECES/SEGUNDO
void Timer0_ISR (void) interrupt 1
{

}

// INSERTE AQUI LA RUTINA DE INTERRUPCION DE LA INTERRUPCION
// 0. SE ACTIVA POR FLANCO DE BAJADA
void INT0_ISR (void) interrupt 0
{
    swc=0;
}

// INSERTE AQUI LA RUTINA DE INTERRUPCION DE LA INTERRUPCION
// 1. SE ACTIVA POR FLANCO DE BAJADA
void INT1_ISR (void) interrupt 2
{
    swb=0;
}

```



9.1.3.- Código fuente del programa para probar la célula de retraso.

```
//-----
// PFC Sergio López Batres.c
//-----
// Copyright (C) 2008 UC3M.
//
// AUTH: LH, SL
// DATE: 3 FEB 2010
//
// Este programa contiene las cabeceras del programa principal,
// la configuracion y las rutinas de interrupcion del 8051F330
//
// Target: C8051F330/1/2/3/4/5
//
// Tool chain: KEIL Eval 'c'
//

//-----
// Includes
//-----
#include "C8051F330.h"
#include <stdio.h>

unsigned int cont_phase=0,cont1=0,cont2=0,cont3=0,cont4=0,ancho=0; //declare aqui
las variables enteras de su programa (sin signo)
unsigned char flag='0',flag1='0';flag2='0',flagprueba='0'; lp=6; //declare aqui las
variables byte o char de su programa (sin signo)

sbit LED = P1^3;
sbit precharge = P1^0;
sbit phi2r = P1^1;
sbit swb = P1^2;
sbit phi1r = P0^6;
sbit cmux = P0^7;
sbit swc = P1^4;
sbit PIN_INT0 = P0^2;
sbit PIN_INT1 = P0^3;

void Init_Device(void);

//-----
// MAIN Routine
// NO TOQUE EL CODIGO DE AQUI HASTA LA SIGUIENTE MARCA
//-----
main (void) {
Init_Device();
TI0=1;
PCA0MD &= ~0x40;
```



// INSERTE EL CODIGO DE SU PROGRAMA PRINCIPAL AQUI. RECUERDE QUE TIENE QUE SER UN BUCLE SIN FIN

```
while (1) {

    if(flag=='0')
    {
        printf("\nProyecto de fin de carrera: Sergio Lopez");
        printf("\nintroduzca la longitud del pulso ");
        lp=getchar()-'0';
        flag='1';
    }

}
```

// FIN DEL PROGRAMA PRINCIPAL

// INSERTE AQUI LA RUTINA DE INTERRUPCION DEL ADC. SE DISPARA AUTOMATICAMENTE

```
void ADC0_ISR (void) interrupt 10
{
    if(cont_phase==0)
    {
        flag1='0'; //Usamos tanto flag 1 como flag 2 para que la
activación de los interruptores sea lo más rápida posible.
        flag2='0';
        if(cont1<2){ // El contador lo ponemos a 8 máximo,
para que la frecuencia de funcionamiento del circuito sea 1khz, ya que 9
interrupciones del timer AC equivalen aproximadamente a 1ms.
            philr=1; //Activamos el pin para borrar el
condensador

            cont1++;
            cmux=0;
        }
        else
        {
            philr=0; //siguiente fase incrementando el
contador de fase.

            cont_phase++;
            cont1=0;
        }
    }
    if(cont_phase==1)
    {
        if(cont2<16)
        {
```


if(flag1=='0')//Haciendo esto conseguimos que la activación del interruptor sea lo más rápida posible para que interfiera lo mínimo en la copia de señales

```

{
    swc=1; //Activamos el tercer circuito de
    interruptores para copiar la señal anterior
    flag1='1';
}
if(flagprueba=='1')
{
    if(cont2<lp)
    {
        precharge=1; //En la fase 1 es cuando se
        ha de cargar el condensador A a la tensión que queramos, ya que durante la fase 0,
        éste se borra, lo cargamos gracias a la señal precharge.
    }
    else
    {
        precharge=0;
    }
}
else
{
    if(cont2<2)
    {
        precharge=1; //En la fase 1 es cuando se
        ha de cargar el condensador A a la tensión que queramos, ya que durante la fase 0,
        éste se borra, lo cargamos gracias a la señal precharge.
    }
    else
    {
        precharge=0;
    }
}

cont2++;
}
else
{
    cont_phase++;
    cont2=0;
}
}
if(cont_phase==2)
{
    if(cont3<2)
    {
        phi2r=1;
        cont3++;
        cmux=1;
    }
}

```



```

    }
    else
    {
        phi2r=0;
        cont_phase++;
        cont3=0;
    }
}
if(cont_phase==3)
{
    if(cont4<16)
    {
        if(flag2=='0')
        {
            swb=1;
            flag2='1';
        }
        cont4++;
    }
    else
    {
        cont4=0;
        cont_phase=0;
        if(flagprueba=='1')
        {
            flagprueba='0';
        }
        else
        {
            flagprueba='1';
        }
    }
}
}

AD0INT=0;
}

```

// INSERTE AQUI LA RUTINA DE INTERRUPCION DEL TIMER 0. SE DISPARA AUTOMATICAMENTE 31 VECES/SEGUNDO

```

void Timer0_ISR (void) interrupt 1
{

}

```

// INSERTE AQUI LA RUTINA DE INTERRUPCION DE LA INTERRUPCION 0. SE ACTIVA POR FLANCO DE BAJADA

```

void INT0_ISR (void) interrupt 0
{
    swc=0;
}

```



// INSERTE AQUI LA RUTINA DE INTERRUPCION DE LA INTERRUPCION

1. SE ACTIVA POR FLANCO DE BAJADA

```
void INT1_ISR (void) interrupt 2
{
  swb=0;
}
```



9.2.- Hojas de características. (Datasheet)

TOOLSTICK C8051F330 DAUGHTER CARD USER'S GUIDE

1. Handling Recommendations

To enable development, the ToolStick Base Adapter and daughter cards are distributed without any protective plastics. To prevent damage to the devices and/or the host PC, please take into consideration the following recommendations when using the ToolStick:

- Never connect or disconnect a daughter card to or from the ToolStick Base Adapter while the Base Adapter is connected to a PC.
- Always connect and disconnect the ToolStick Base Adapter from the PC by holding the edges of the boards.

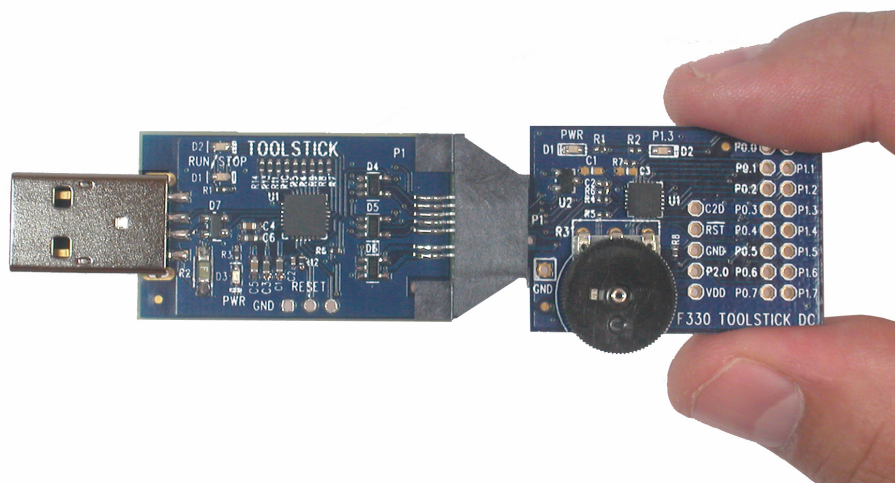


Figure 1. Proper Method of Holding the ToolStick

- Avoid directly touching any of the other components.

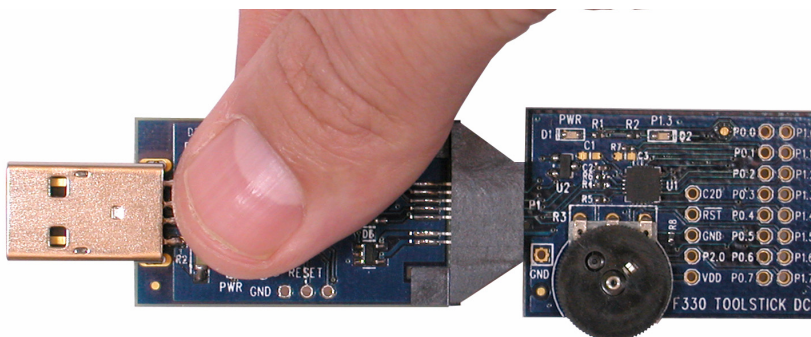


Figure 2. Improper Method of Holding the ToolStick

- Manipulate mechanical devices on the daughter cards, such as potentiometers, with care to prevent the Base Adapter or daughter card from accidentally dislodging from their sockets.

ToolStick-F330DC

2. Contents

The ToolStick-F330DC kit contains the following items:

- ToolStick C8051F330 Daughter Card

The ToolStick Starter Kit includes the following items:

- ToolStick Base Adapter
- ToolStick C8051F330 Daughter Card
- 3-foot USB extension cable

A ToolStick daughter card requires a ToolStick Base Adapter to communicate with the PC. If the daughter card was not purchased as part of a Starter Kit, ToolStick Base Adapters can be purchased separately at www.silabs.com/toolstick.

3. ToolStick Overview

The purpose of the ToolStick is to provide a development and demonstration platform for Silicon Laboratories microcontrollers and to demonstrate the Silicon Laboratories software tools, including the Integrated Development Environment (IDE).

The ToolStick development platform consists of two components: the ToolStick Base Adapter and a daughter card. The ToolStick Base Adapter provides a USB debug interface and data communications path between a Windows PC and a target microcontroller.

The target microcontroller and application circuitry are located on the daughter card. Some daughter cards, such as the C8051F330 Daughter Card, are used as general-purpose development platforms for the target microcontrollers and some are used to demonstrate a specific feature or application.

The C8051F330 Daughter Card includes a pair of LEDs, a potentiometer, a resistor across the C8051F330's current DAC output pin, and a small prototyping area which provides access to all of the pins of the device. This prototyping area can be used to connect additional hardware to the microcontroller and use the daughter card as a development platform. See Section "10. Board Revision Information," on page 14 for information regarding the prototyping area.

Figure 3 shows the ToolStick C8051F330 Daughter Card and identifies the various components.

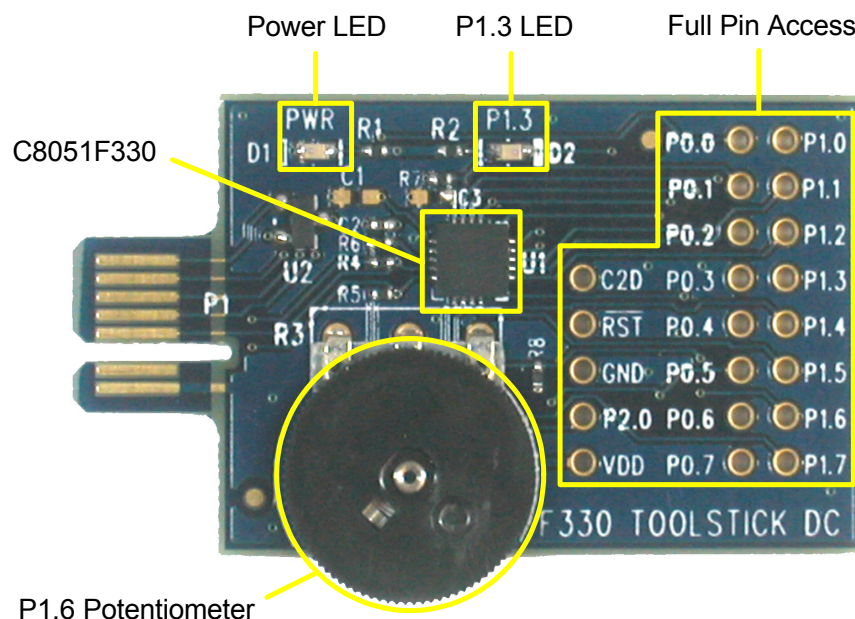


Figure 3. ToolStick C8051F330 Daughter Card

4. Getting Started

The necessary software to download, debug and communicate with the target microcontroller must be downloaded from www.silabs.com/toolstick. The following software is necessary to build a project, download code to, and communicate with the target microcontroller:

- Silicon Laboratories Integrated Development Environment (IDE)
- Keil Demonstration Tools
- ToolStick Terminal application

The Keil Demo Tools include a compiler, assembler, and linker. The limits of the demo version are: 1) the resulting object code is limited to 2 kBytes and 2) the floating point library is not included. ToolStick Terminal communicates with the target microcontroller's UART through the ToolStick Base Adapter. It can also read/write the two GPIO pins available on the ToolStick Base Adapter.

Other useful software that is provided on the Silicon Labs Downloads (www.silabs.com/mcudownloads) website includes:

- Configuration Wizard 2
- Keil uVision2 and uVision3 Drivers

All of the above software is described in more detail in Section “5. Software Overview”.

To simplify the download process, the necessary software described above is provided in two download packages. The ToolStick Download package includes the Keil Tools, example code, documentation including User's Guides and data sheets, and the ToolStick Terminal application. The IDE, Configuration Wizard 2, and the Keil uVision Drivers are available as a separate download. After downloading and installing these two packages, see the following sections for information regarding the software and running one of the demo applications.

5. Software Overview

5.1. Silicon Laboratories IDE

The Silicon Laboratories IDE integrates a source-code editor, source-level debugger, and an in-system Flash programmer. See Section “6. ToolStick C8051F330 Daughter Card Features Demo” for detailed information on how to use the IDE. The Keil Demonstration Toolset includes a compiler, linker, and assembler and easily integrates into the IDE. The use of third-party compilers and assemblers is also supported.

5.1.1. IDE System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 2000 or Windows XP.
- One available USB port.
- 64 MB RAM and 40 MB free HD space recommended.

5.1.2. 3rd Party Toolsets

The Silicon Laboratories IDE has native support for many 8051 compilers. The full list of natively supported tools is:

- Keil
- IAR
- Raisonance
- Tasking
- Hi-Tech
- SDCC
- Dunfield

Please note that the demo applications for the C8051F330 Daughter Card are written for the Keil toolset.

ToolStick-F330DC

5.2. Keil Demonstration Toolset

5.2.1. Keil Assembler and Linker

The assembler and linker that are part of the Keil Demonstration Toolset are the same versions that are found in the full Keil Toolset. The complete assembler and linker reference manual can be found on-line under the **Help** menu in the IDE or in the “*SiLabs\MCU\hlp*” directory (A51.pdf).

5.2.2. Keil Demonstration C51 C Compiler

The demonstration version of the C51 compiler is the same as the full version except code size is limited to 2 kB and the floating point library is not included. The C51 compiler reference manual can be found under the **Help** menu in the IDE or in the “*SiLabs\MCU\hlp*” directory (C51.pdf).

5.3. Configuration Wizard 2

The Configuration Wizard 2 is a code generation tool for all of the Silicon Laboratories devices. Code is generated through the use of dialog boxes for each of the device's peripherals.

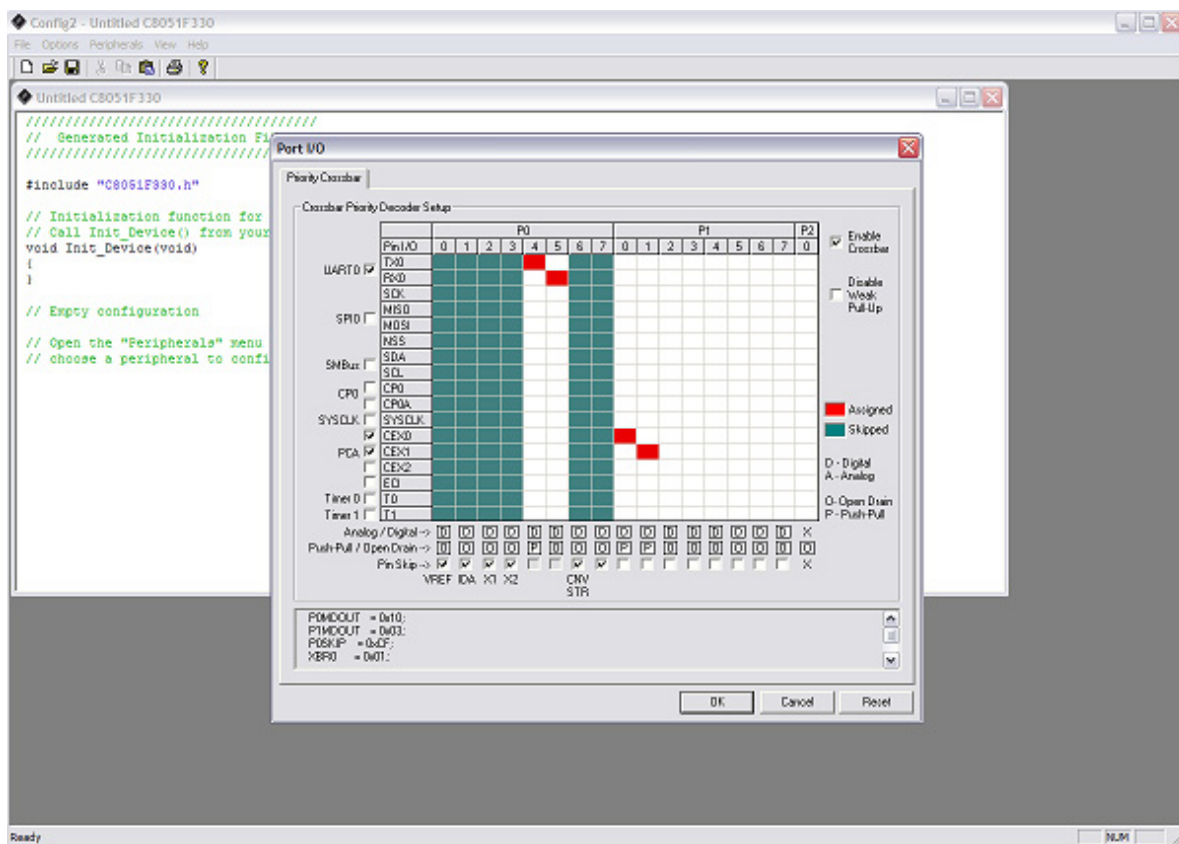


Figure 4. Configuration Wizard 2 Utility

The Configuration Wizard 2 utility helps accelerate development by automatically generating initialization source code to configure and enable the on-chip resources needed by most design projects. In just a few steps, the wizard creates complete startup code for a specific Silicon Laboratories MCU. The program is configurable to provide the output in C or assembly.

For more information, please refer to the Configuration Wizard 2 documentation. The documentation and software available from the Downloads webpage (www.silabs.com/mcudownloads).

5.4. Keil uVision2 and uVision3 Silicon Laboratories Drivers

As an alternative to the Silicon Laboratories IDE, the uVision debug driver allows the Keil uVision2 and uVision3 IDEs to communicate with Silicon Laboratories on-chip debug logic. In-system Flash memory programming integrated into the driver allows for rapidly updating target code. The uVision2 and uVision3 IDEs can be used to start and stop program execution, set breakpoints, check variables, inspect and modify memory contents, and single-step through programs running on the actual target hardware.

For more information, please refer to the uVision driver documentation. The documentation and software are available from the Downloads webpage (www.silabs.com/mcudownloads).

5.5. ToolStick Terminal

The ToolStick Terminal program provides the standard terminal interface to the target microcontroller's UART. However, instead of requiring the usual RS-232 and COM port connection, ToolStick Terminal uses the USB interface of the ToolStick Base Adapter to provide the same functionality.

In addition to the standard terminal functions (send file, receive file, change baud rate), two GPIO pins on the target microcontroller can be controlled using the Terminal for either RTS/CTS handshaking or software-configurable purposes (see the demo software for an example).

See Section 6.8 for more information on using ToolStick Terminal. The software is available on the ToolStick webpage (www.silabs.com/toolstick).

ToolStick-F330DC

6. ToolStick C8051F330 Daughter Card Features Demo

The ToolStick kit includes a few simple code examples. The example described in this section is titled **F330DC_FeaturesDemo**. The purpose of this example is to guide a new user through the features and capabilities of the IDE and demonstrate the microcontroller's on-chip debug capabilities. The **F330DC_FeaturesDemo** example code uses the potentiometer on the daughter card to vary the blinking rate of the LED. The first part of this demo shows how to use the IDE to connect and download the firmware, view and modify registers, use watch windows, use breakpoints, and single step through code. The second part of the demo shows how to use ToolStick Terminal to receive UART data from the daughter card and how to use the GPIO pins.

6.1. Hardware Setup

Connect the ToolStick hardware to the PC using the steps below while taking note of the recommendations in Section 1:

1. Connect the ToolStick Base Adapter to the ToolStick C8051F330 Daughter Card.
2. If available, connect the USB extension cable to the ToolStick Base Adapter.
3. Connect the ToolStick to a USB port on a PC.

See Figure 5 below for an example hardware setup.

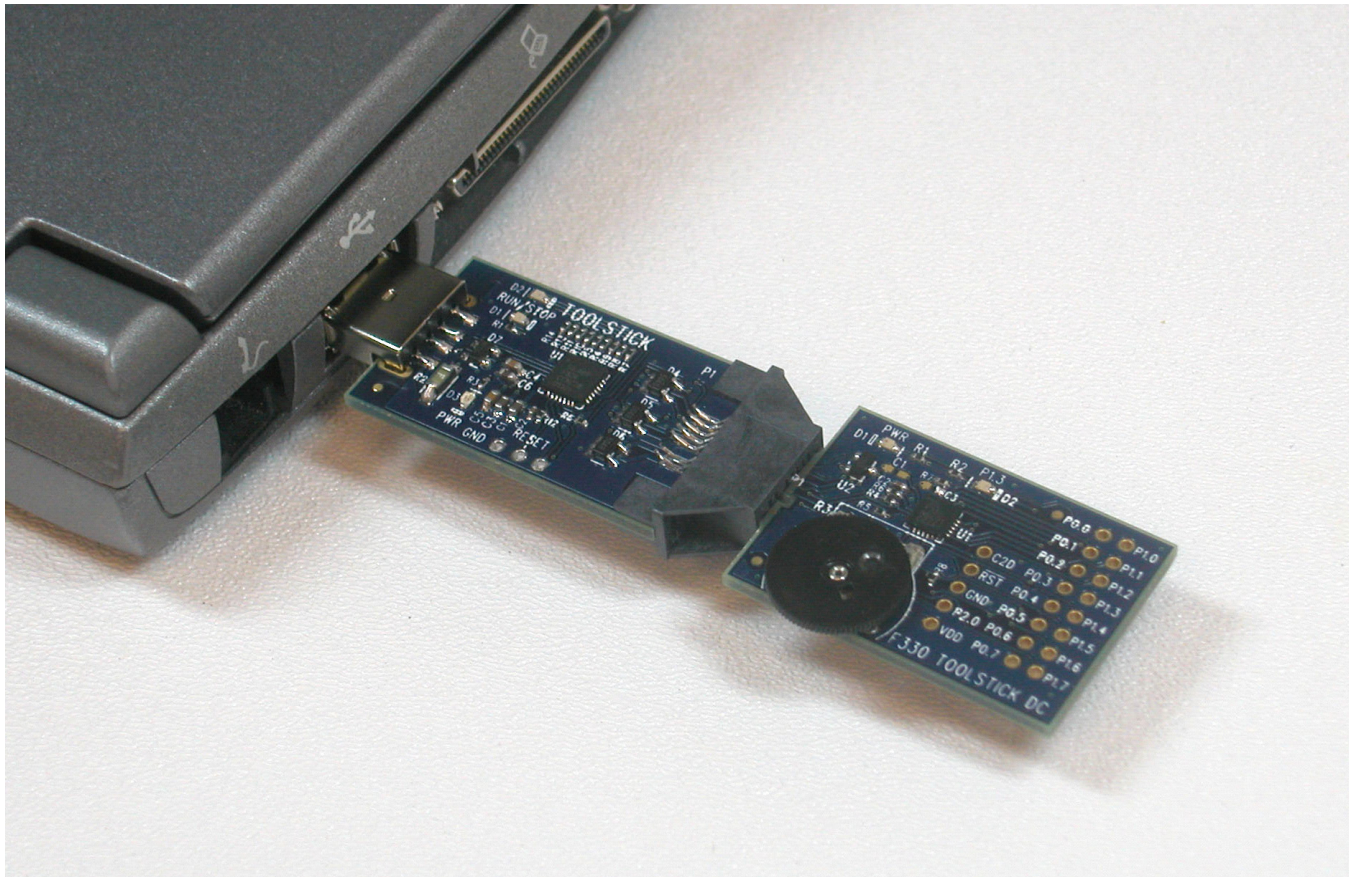


Figure 5. Hardware Setup Example

6.2. Connecting to the Device and Downloading Firmware

This section describes how to open the IDE, open and build a project, connect to a device and download the firmware.

1. Open the Silicon Laboratories IDE from the **Start** → **Programs** → **Silicon Laboratories** menu.
2. In the IDE, go to **Project** → **Open Project**.
3. Browse to `C:\SiLabs\MCU\ToolStick\F330DC\Firmware\`.
4. Select **F330DC_FeaturesDemo.wsp** and click OK.
5. In the IDE, select **Project** → **Rebuild Project**.
6. Go to **Options** → **Connection Options**.
7. Select “**USB Debug Adapter**” for the Serial Adapter and “**C2**” for the Debug Interface, and then click “OK”.
8. Go to **Debug** → **Connect**.
9. Download the code using the **download button** on the menu bar or use alt-D.

Once these steps are completed, the firmware is built into an object file (step 5) and downloaded to the device (step 9). The device is now ready to begin executing code. If all of these steps were followed successfully, the “Go” option is enabled in the Debug menu. A green circle icon in the IDE toolbar also indicates that the device is ready to run. If one of the steps leads to an error, make sure that the ToolStick is properly inserted in a USB port and start again with step 6.

6.3. Running and Stopping Code Execution

Once the IDE is connected to the device and the firmware is loaded, the IDE can start and stop the code execution. The following steps can be performed using the buttons on the toolbar or using the options in the Debug menu.

1. To start code execution, click the green “**Go**” button on the toolbar or use the **Debug** → **Go** menu option. The green LED on the daughter card will start to flash. The debug commands in the IDE (single-step, multiple-step, set breakpoint, and others) are disabled when the device is running. While the firmware is running, the potentiometer on the daughter card can be turned to alter the blinking speed of the LED.



2. To stop code execution, click the red “**Stop**” button on the toolbar or use the **Debug** → **Stop** menu option. The device will halt code execution and all of the registers and pins on the device will hold their state.



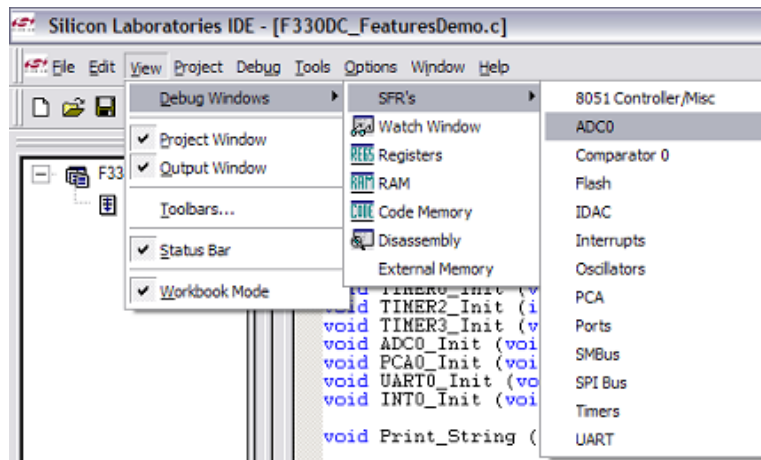
All debug windows and watch windows are refreshed when the device is stopped. If any of the values in these windows have changed since the last time the device was halted, the new value is shown in red text instead of black text.

ToolStick-F330DC

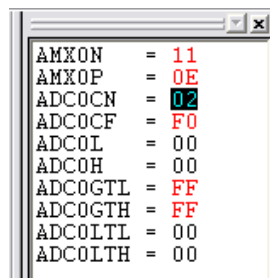
6.4. Viewing and Modifying Registers

All registers on the device can be viewed and modified when the device is in a halted state. The registers are grouped together according to which peripheral or part of hardware they belong. As an example, this guide shows how to open the ADC0 Debug Window and disable the ADC0 directly from the IDE.

1. Open the ADC0 Debug Window from the **View** → **Debug Windows** → **SFR's** → **ADC0** menu option. The ADC0 Debug Window appears on the right-hand side of the IDE. In this window, the ADC0CN register is shown. This register is used to enable and configure the on-chip ADC. When the firmware is running, the ADC0CN register reads as 0x82 indicating that the ADC is running.



2. In the debug window, change the value of **ADC0CN** from 0x82 to 0x02. This value turns off the ADC on the target microcontroller.



3. To write this new value to the device, select **Refresh** from the Debug Menu or click the **Refresh** button in the toolbar.



4. Click "Go" to resume running the device with the new **ADC0CN** value.
5. Turn the potentiometer on the daughter card and notice that it has no effect on the blinking rate of the LED.
6. Re-enable the ADC by writing 0x82 to the **ADC0CN** and clicking the **Refresh** button.

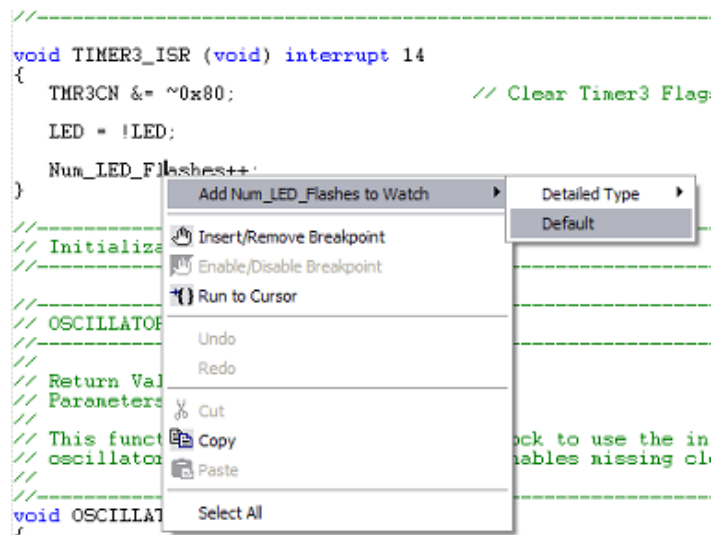
Changing the values of registers does not require recompiling the code and redownloading the firmware. At any time, the device can be halted and the values of the registers can be changed. After selecting "Go", the firmware will continue execution using the new values. This capability greatly speeds up the debugging process. See the data sheet for the C8051F330 device for the definitions and usage for all registers.

The debug windows for other sets of registers are found in the **View** → **Debug Windows** → **SFR's** menu.

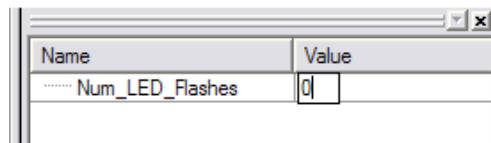
6.5. Enabling and Using Watch Windows

The Debug Windows in the View menu are used to view and modify hardware registers. To view and modify variables in code, the IDE provides Watch Windows. Just as with register debug windows, variables in the watch windows are updated each time the device is halted. This section of the User's Guide explains how to add a variable to the watch window and modify the variable. In the F330_FeatureDemo example code, the variable **Num_LED_Flashes** is a counter that stores the number of times the LED blinks.

1. If the device is running, stop execution using the **Stop** button or use the **Debug** → **Stop** menu option.
2. In the File View on the left-hand side of the IDE, double-click on **F330DC_FeaturesDemo.c** to open the source file.
3. Scroll to the **Timer3_ISR** function (line 334) and right-click on the variable **Num_LED_Flashes**. In the context menu that appears, select **Add Num_LED_Flashes to Watch** and then choose **Default.** On the right-hand portion of the IDE, the watch window appears and the variable is added. The current value of the variable is shown to the right of the name.



4. **Start** and **stop** the device a few times. See that the value of the **Num_LED_Flashes** is incremented each time the LED blinks.
5. When the device is halted, click on the value field in the watch window and change the value to 0. Then click the **Refresh** button or select **Debug** → **Refresh** to write the new value to the device.



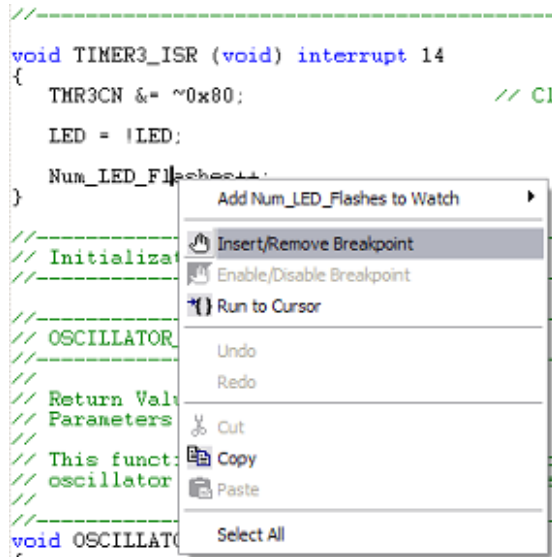
6. **Start** and **stop** the device a few times to watch the variable increment starting at 0.

Changing the values of variables does not require recompiling the code and redownloading the firmware. At any time, the device can be halted and the values of the variables can be changed. The firmware will continue execution using the new values.

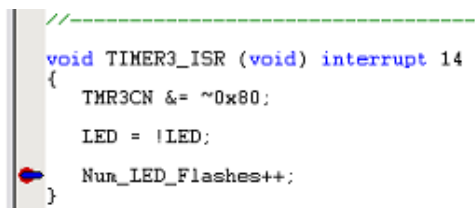
6.6. Setting and Running to Breakpoints

The Silicon Laboratories microcontroller devices support up to four hardware breakpoints. A breakpoint is associated with a specific line of code. When the processor reaches a hardware breakpoint, the code execution stops, and the IDE refreshes all debug and watch windows. The on-chip debug hardware allows for breakpoints to be placed on any line of executable code, including code in Interrupt Service Routines. This section provides steps to set a breakpoint on the line of source code that increments the **Num_LED_Flashes** variable.

1. If the device is running, stop execution using the **“Stop”** button or use the **Debug → Stop** menu option.
2. Scroll to the **Timer3_ISR** function (line 334) and right-click on the variable **“Num_LED_Flashes”**. In the context menu that appears, select **“Insert/Remove Breakpoint.”** On the left side of the line in the editor window, a red circle is added to indicate a breakpoint is placed on the source line.



3. Click the **“Go”** button or select the **Debug → Go** menu option.
4. After a short time, the IDE will show that the device is halted. A blue line will be placed in the editor window to indicate where the code execution has stopped.

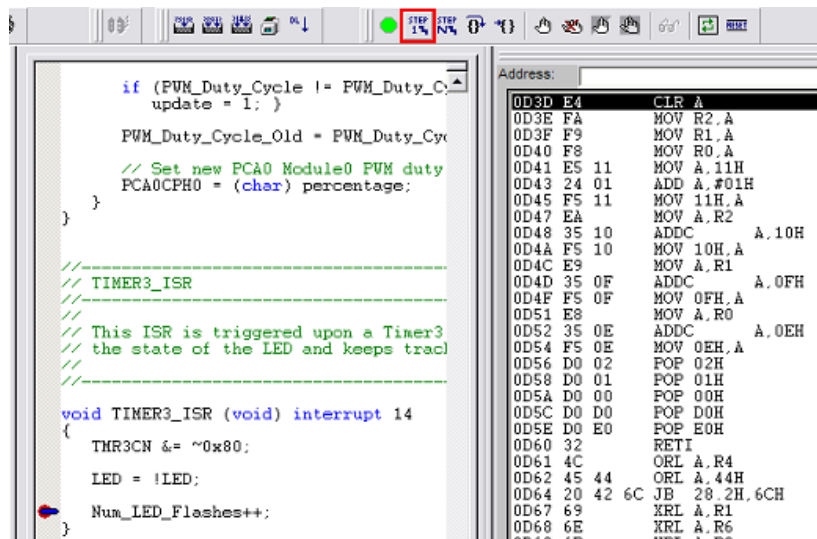


5. **Start** and **stop** the processor a few more times. Notice that the LED blinks once for every time the processor is started and the **Num_LED_Flashes** variable also increments by one.

6.7. Single-Stepping Through Firmware

The IDE supports the ability to single-step through firmware one assembly instruction at a time. The IDE reads the Flash from the device, converts the instructions to assembly and displays them in a disassembly window. The following steps show how to open the disassembly window and single step through firmware.

1. If there is already not a breakpoint set on line of code that increments the **Num_LED_Flashes** variable, set the breakpoint using the steps described in Section 6.6.
2. Start the processor using the “Go” button and wait till it stops on the breakpoint.
3. Select **View** → **Debug Windows** → **Disassembly**. The disassembly window will appear on the right-hand side of the IDE, if it is not already open.
4. To execute one assembly instruction at a time, click the “Step” button on the toolbar or select the **Debug** → **Step** menu option. The highlighted line in the disassembly window indicates the next instruction to be executed. The blue line marker in the editor window will stay on the same .C source line until all of the assembly instructions are completed.



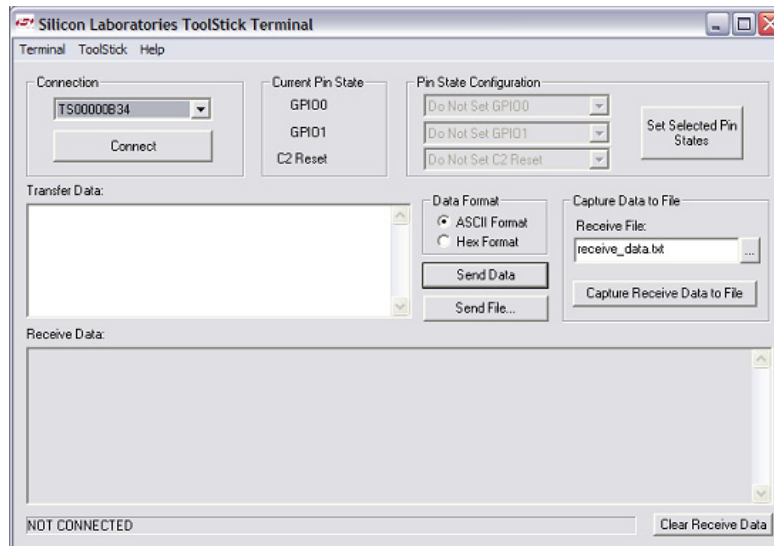
The disassembly window has three columns. The left column is the address of the instruction in Flash. The middle column is the instruction in hex. The right column is the disassembled instruction. The Disassembly debug window and the capability to single-step through firmware allows a developer to see exactly what instructions are executed and their output.

ToolStick-F330DC

6.8. Using ToolStick Terminal

This section describes how to use ToolStick Terminal to communicate with UART from the PC to the daughter card through the ToolStick Base Adapter.

1. If the Silicon Laboratories IDE is open, close the IDE. The IDE and the ToolStick Terminal cannot communicate with the daughter card simultaneously.
2. Open ToolStick Terminal from the **Start** → **Programs** → **Silicon Laboratories** menu.



3. Go to the **ToolStick** → **Settings** menu.
4. Under “Pin Settings”, change GPIO0 / RTS to “**GPIO Output - Push Pull**” and click “OK.” The rest of the default settings are correct for the C8051F330 Features Demo.
5. In the top, left-hand corner of the Terminal application, available devices are shown in the drop-down Connection menu. Click “**Connect**” to connect to the device. In the “**Receive Data**” window, text indicating the blink rate of the LED will appear.
6. Turn the potentiometer on the daughter card and see that the blink rate is updated on the daughter card and the new blink rate is printed to the Terminal.

In addition to the standard two UART pins (TX and RX), there are two GPIO/UART handshaking pins on the ToolStick Base Adapter that are connected to two port pins on the target microcontroller. ToolStick Terminal is used to configure and read/write these pins. For the **F330DC_FeaturesDemo**, one of these GPIO pins is connected to an external interrupt pin on the C8051F330. The following steps describe how to change the level of one of the GPIO pins and trigger an interrupt on the target microcontroller. The interrupt forces the firmware to switch modes and send a pulse-width modulated (PWM) signal to the LED instead of blinking the LED using an on-chip Timer.

1. In ToolStick Terminal, under Pin State Configuration, select “**Set GPIO0 Logic Low**” and click on “**Set Selected Pin States.**” This changes the level of the GPIO0 pin from **Logic High** to **Logic Low** and triggers a level-sensitive interrupt on the microcontroller.
2. In the Receive window, see that the printed text has changed to indicate the LED PWM duty cycle.
3. Turn the potentiometer on the daughter card to change the brightness of the LED on the daughter card.
4. Change the GPIO0 pin state back to **Logic High** and notice that the firmware switches back to blinking the LED.

The firmware on the C8051F330 target microcontroller does not need to be customized to use the UART and communicate with ToolStick Terminal. The firmware on the microcontroller should write to the UART as it would in any standard application and all of the translation is handled by the ToolStick Base Adapter.

7. Additional Demo Examples

In addition to the **F330DC_FeaturesDemo** example firmware, the ToolStick download package also includes demo projects named **F330DC_ADC0_TemperatureSensor** and **F330DC_IDA0_SineWave**. The instructions for running these demos can be found at the top of the source file.

The project and source files for these demos can be found in the *C:\SiLabs\MCU\ToolStick\F330DC\Firmware* folder.

8. Using the C8051F330 Daughter Card as a Development Platform

The prototyping area on the ToolStick C8051F330 daughter card makes it easy to interface to external hardware. All of the digital I/O pins are available so it possible to create a complete system.

8.1. C8051F330 Pin Connections

It is important to note that if external hardware is being added, some of the existing components on the board can interfere with the signaling. The following is a list of port pins on the C8051F330 that are connected to other components:

- P0.1—This pin has a pull-down resistor (R7) to ground so that the IDAC output can be converted to a voltage. R7 can be safely removed from the daughter card if it is not needed.
- P0.4, P0.5—These pins are connected directly to the ToolStick Base Adapter for UART communication.
- P0.6, P0.7—These pins are connected directly to the ToolStick Base Adapter's GPIO pins. By default, these GPIO pins on the Base Adapter are high-impedance pins so they will not affect any signaling. Configuring these pins on the Base Adapter to output pin or handshaking pins could affect signaling.
- P1.3—This pin is connected to the cathode of the green LED on the daughter card. The LED or the R2 resistor can be removed to disconnect the LED from the pin.
- P1.6—This pin is connected to the output of the potentiometer. The 0 ohm resistor can be removed to disconnect the potentiometer from the pin.

See the daughter card schematic in Section 11 for more information.

8.2. VREF Capacitor

On the C8051F330 devices, if VREF is generated internally, it is output to port pin P0.0. For VREF stability, it is highly recommended to place a capacitor on the VREF output pin. On the ToolStick C8051F330 Daughter Card, there are pads on the board (C3) to populate a 0603 surface mount capacitor. The firmware examples for the daughter card use VDD as VREF, so no external capacitor on P0.0 is necessary for proper operation.

8.3. C2 Pin Sharing

On the C8051F330, the debug pins, C2CK, and C2D, are shared with the pins /RST and P2.0 respectively. The daughter card includes the resistors necessary to enable pin sharing which allow the /RST and P2.0 pins to be used normally while simultaneously debugging the device. See Application Note "AN124: Pin Sharing Techniques for the C2 Interface" at www.silabs.com for more information regarding pin sharing.

9. Information Locations

Example source code is installed by default in the “C:\SiLabs\MCU\ToolStick\F330DC\Firmware” directory during the ToolStick installation.

Documentation for the ToolStick kit, including this User’s Guide, can be found in the C:\SiLabs\MCU\ToolStick\Documentation and the C:\SiLabs\MCU\ToolStick\F330DC\Documentation directories.

The installer for the ToolStick software is available at www.silabs.com/toolstick.

10. Board Revision Information

Revision SA-TS002-001 of the F330 Daughter Card has two test points swapped in the prototyping area. The test point labeled P0.3 is connected to pin P1.3, and the test point labeled P1.3 is connected to pin P0.3 on the microcontroller. This error is fixed on revision SA-TS002-002 and later boards. The revision number is located on the back side of the daughter card.

Rev. 0.3

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Added Section "10. Board Revision Information," on page 14.

Revision 0.2 to Revision 0.3

- Added ToolStick Terminal, Configuration Wizard 2, and Keil uVision drivers to Section "5. Software Overview".
- Added images to Sections "6.3. Running and Stopping Code Execution", "6.4. Viewing and Modifying Registers", "6.5. Enabling and Using Watch Windows", "6.6. Setting and Running to Breakpoints", "6.7. Single-Stepping Through Firmware", and "6.8. Using ToolStick Terminal".
- Various small text edits.

NOTES:

ToolStick-F330DC

CONTACT INFORMATION

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032
Email: MCUinfo@silabs.com
Internet: www.silabs.com

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.



VOLTAGE COMPARATORS

- MAXIMUM INPUT CURRENT : 150nA
- MAXIMUM OFFSET CURRENT : 20nA
- DIFFERENTIAL INPUT VOLTAGE RANGE : $\pm 30V$
- POWER CONSUMPTION : 135mW AT $\pm 15V$
- SUPPLY VOLTAGE : +5V TO $\pm 15V$
- OUTPUT CURRENT : 50mA

DESCRIPTION

The LM111, LM211, LM311 are voltage comparators that have low input currents.

They are also designed to operate over a wide range of supply voltages : from standard $\pm 15V$ operational amplifier supplies down to the single +5V supply used for IC logic.

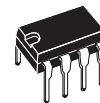
Their output is compatible with RTL-DTL and TTL as well as MOS circuits and can switch voltages up to +50V at outputs currents as high as 50mA.

ORDER CODE

Part Number	Temperature Range	Package	
		N	D
LM111	-55°C, +125°C	•	•
LM211	-40°C, +105°C	•	•
LM311	0°C, +70°C	•	•
Example : LM311D			

N = Dual in Line Package (DIP)

D = Small Outline Package (SO) - also available in Tape & Reel (DT)

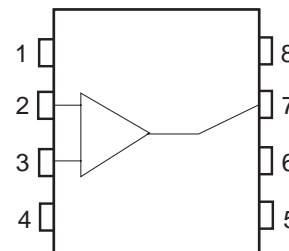


N
DIP8



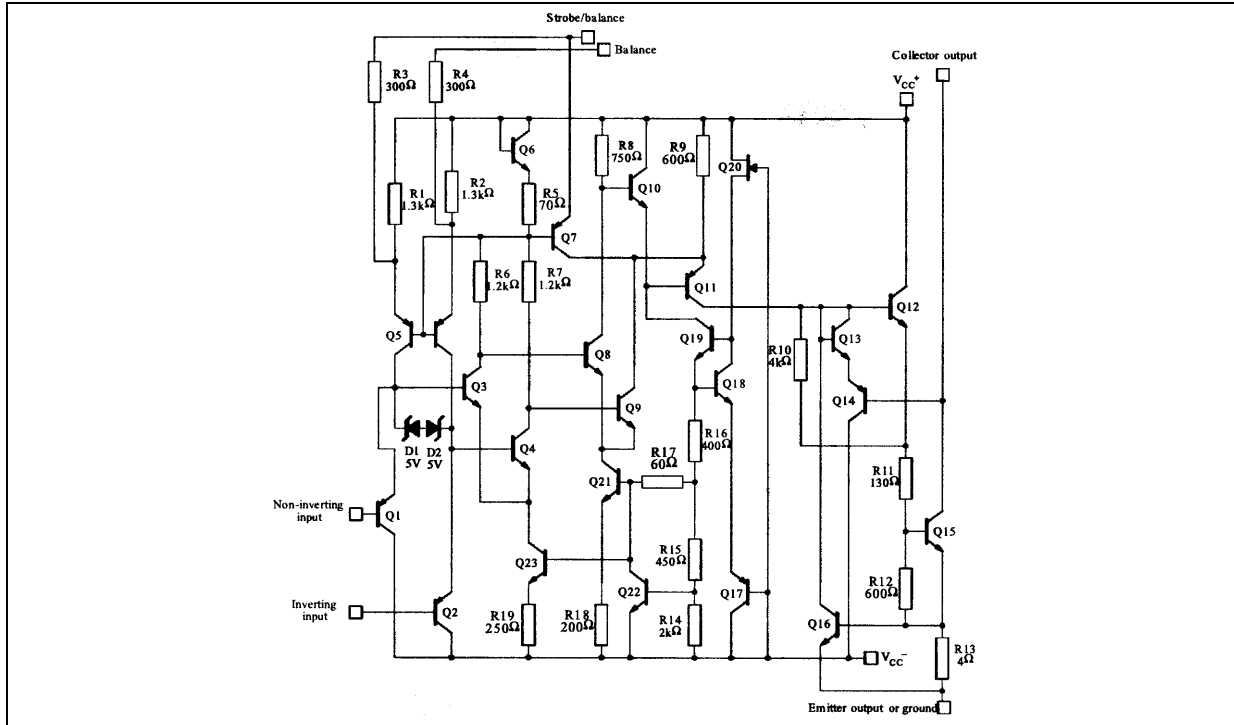
D
SO8

PIN CONNECTIONS (top view)



- | | |
|-------------------------|--------------------|
| 1 - Ground | 5 - Balance |
| 2 - Non-inverting input | 6 - Strobe/Balance |
| 3 - Inverting input | 7 - Output |
| 4 - V_{CC}^- | 8 - V_{CC}^+ |

SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

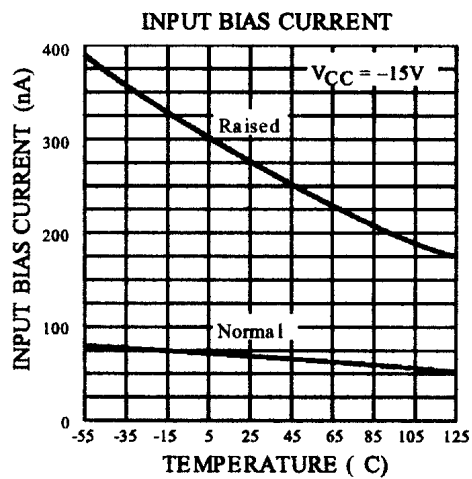
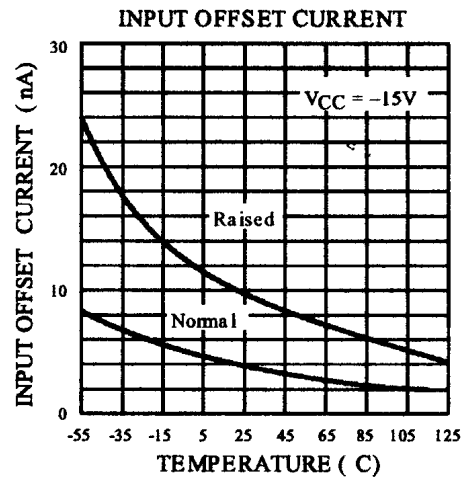
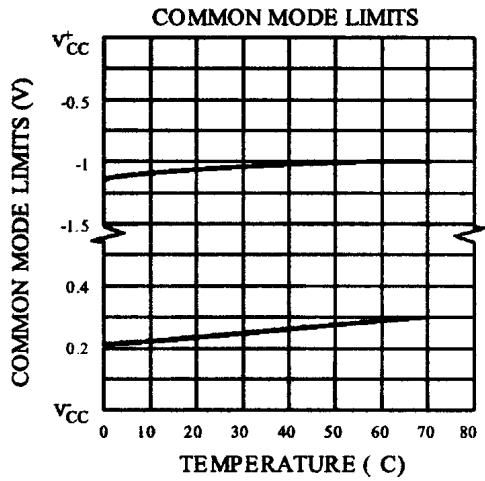
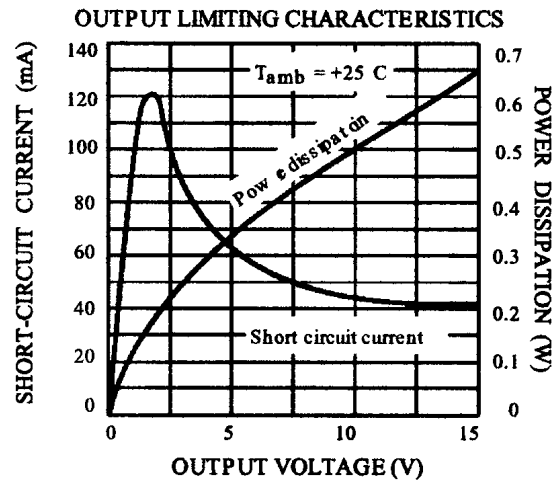
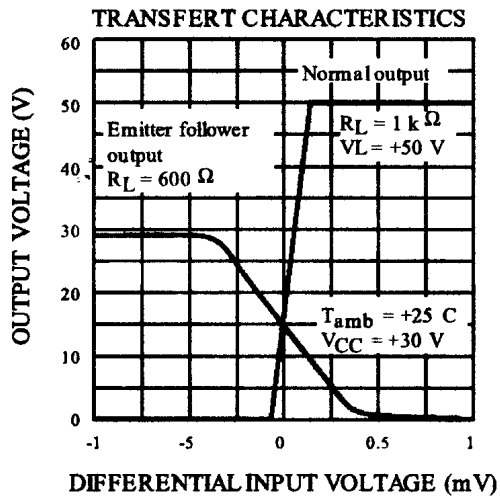
Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	36	V
V_{id}	Differential Input Voltage	± 30	V
V_i	Input Voltage ¹⁾	± 15	V
$V_{(1-4)}$	Ground to Negative Supply Voltage	30	V
$V_{(7-4)}$	Output to Negative Supply Voltage	LM111-LM211 LM311	V
	Output Short-Circuit Duration	10	s
	Voltage at strobe pin	$V_{CC}^+ - 5$	V
P_d	Power Dissipation ²⁾	DIP8 SO8	mW
T_j	Junction Temperature	+150	°C
T_{stg}	Storage Temperature Range	-65 to +150	°C

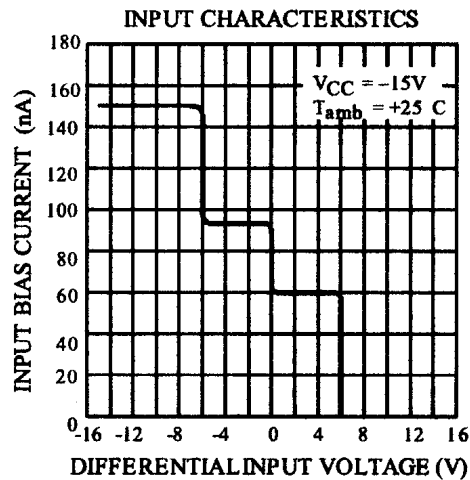
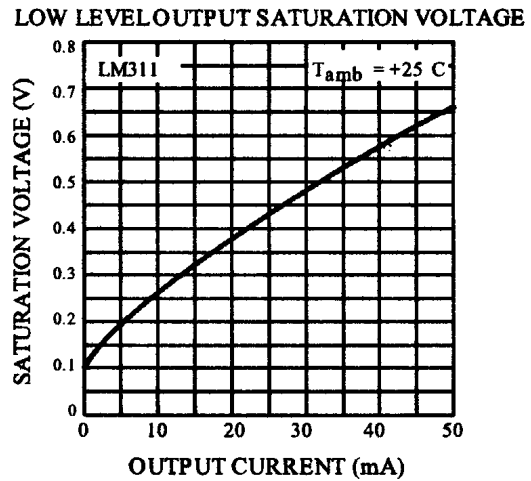
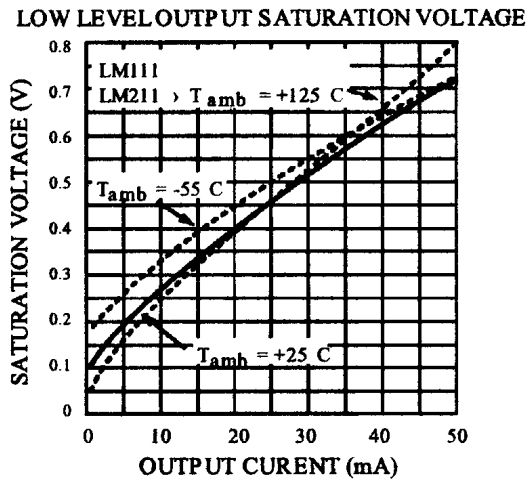
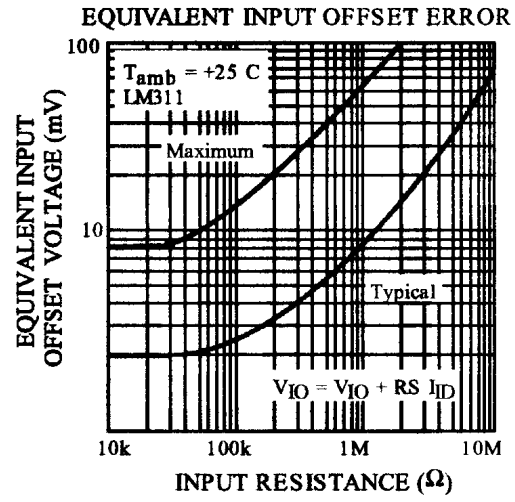
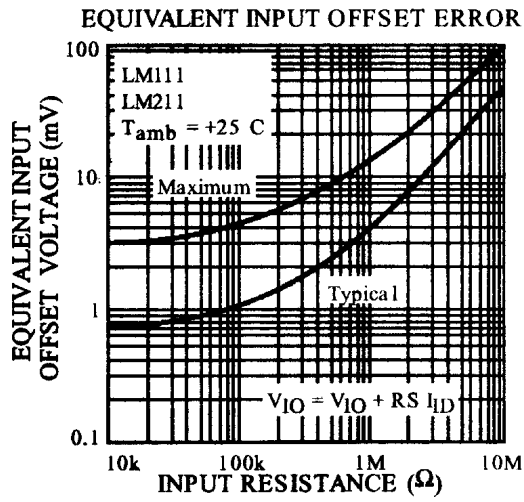
1. This rating applies for $\pm 15V$ supplies. The positive input voltage limit is 30V above the negative. The negative input voltage is equal to the negative supply voltage or 30V below the positive supply, whichever is less.

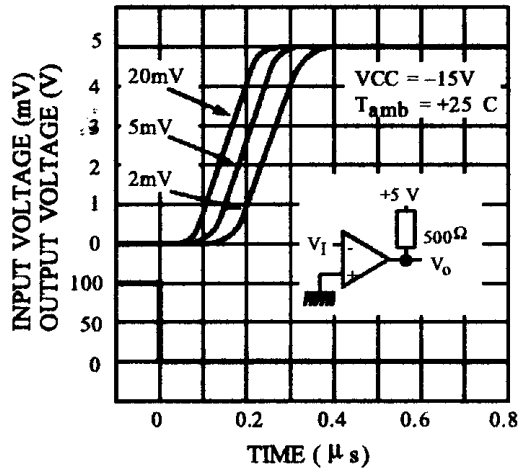
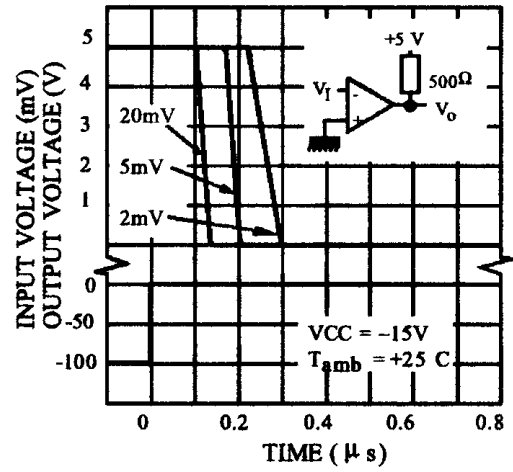
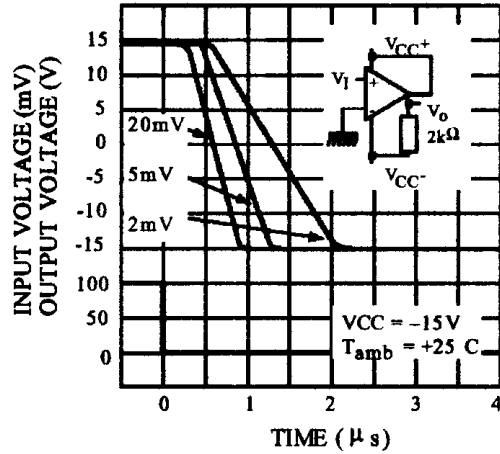
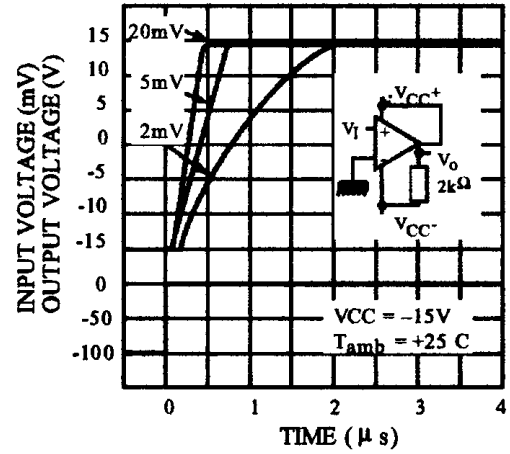
2. P_d is calculated with $T_{amb} = +25^\circ C$, $T_j = +150^\circ C$ and $R_{thja} = 100^\circ C/W$ for DIP8 package
 $R_{thja} = 175^\circ C/W$ for SO8 package

OPERATING CONDITIONS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	5 to ± 15	V
T_{oper}	Operating Free-Air Temperature range	LM111 LM211 LM311	°C

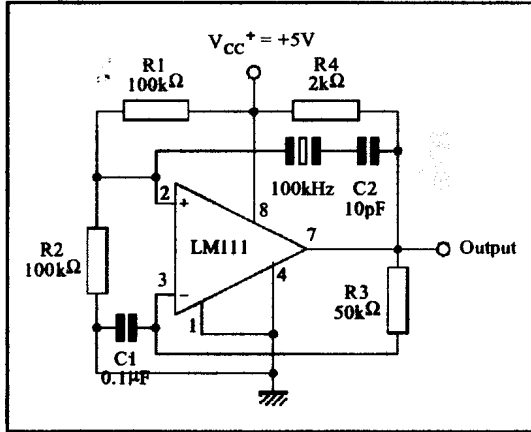




RESPONSE TIME FOR VARIOUS
INPUT OVERDRIVESRESPONSE TIME FOR VARIOUS
INPUT OVERDRIVESRESPONSE TIME FOR VARIOUS
INPUT OVERDRIVESRESPONSE TIME FOR VARIOUS
INPUT OVERDRIVES

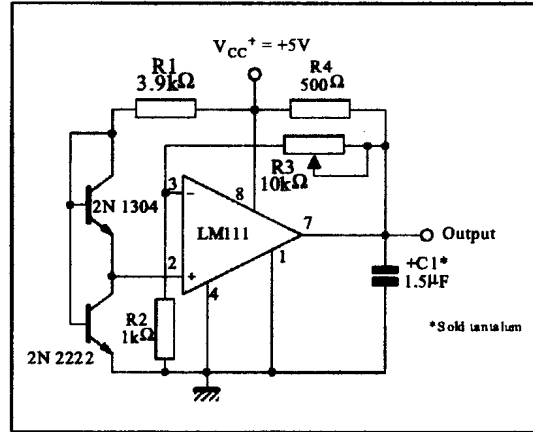
TYPICAL APPLICATIONS

CRYSTAL OSCILLATOR

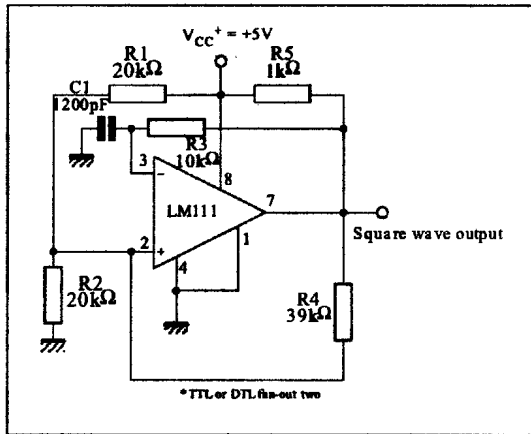


100KHz FREE RUNNING MULTIVIBRATOR

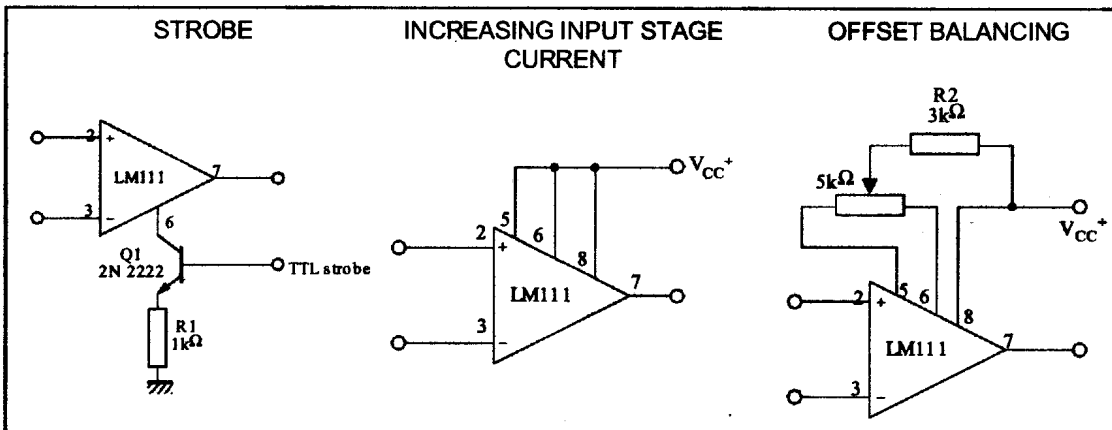
LOW VOLTAGE ADJUSTABLE REFERENCE SUPPLY

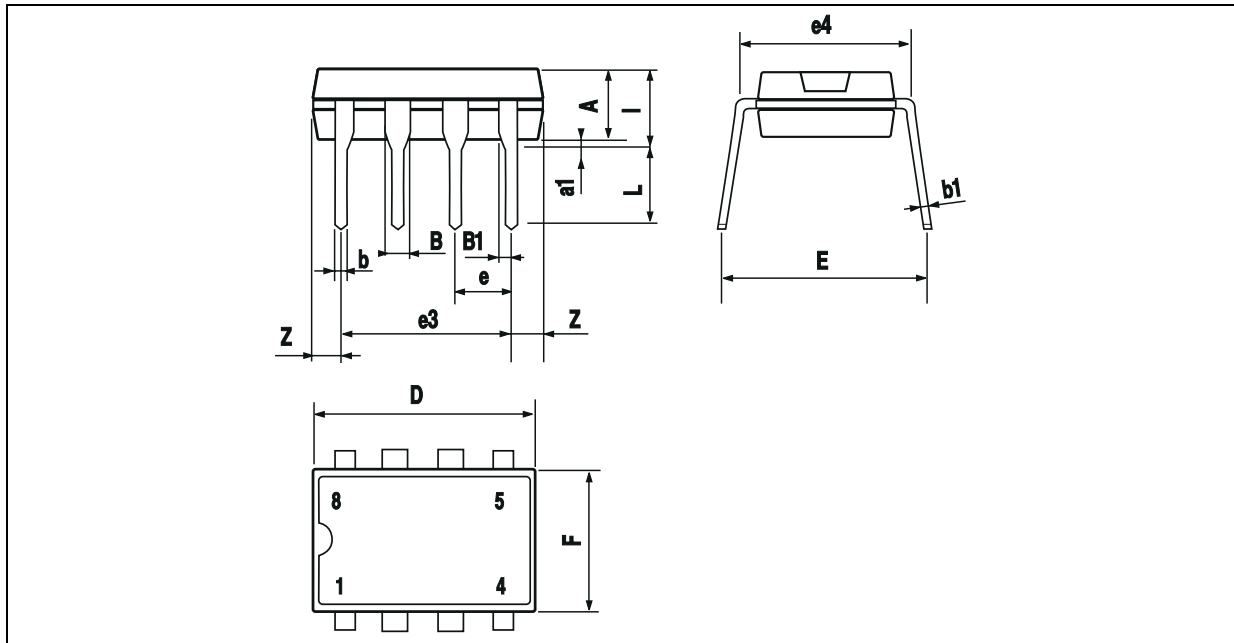


TTL INTERFACE WITH HIGH LEVEL LOGIC



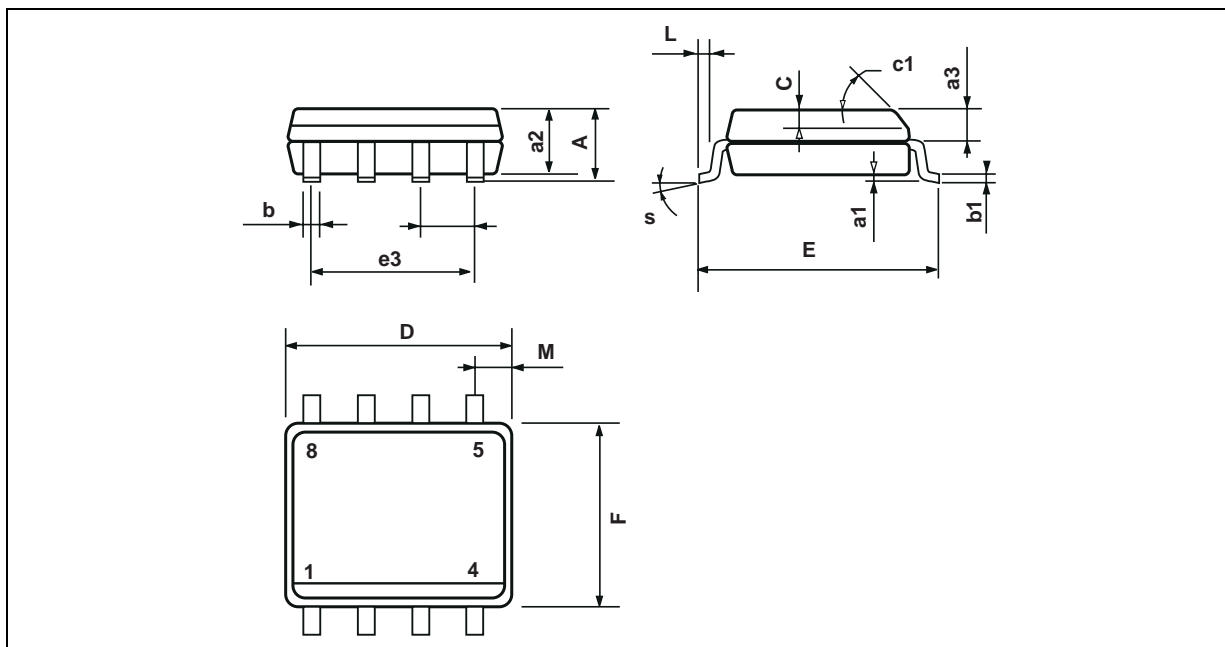
AUXILIARY CIRCUITS



PACKAGE MECHANICAL DATA
 8 PINS - PLASTIC DIP


Dimensions	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A		3.32			0.131	
a1	0.51			0.020		
B	1.15		1.65	0.045		0.065
b	0.356		0.55	0.014		0.022
b1	0.204		0.304	0.008		0.012
D			10.92			0.430
E	7.95		9.75	0.313		0.384
e		2.54			0.100	
e3		7.62			0.300	
e4		7.62			0.300	
F			6.6			0.260
i			5.08			0.200
L	3.18		3.81	0.125		0.150
Z			1.52			0.060

PACKAGE MECHANICAL DATA **8 PINS - PLASTIC MICROPACKAGE (SO)**



Dimensions	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			1.75			0.069
a1	0.1		0.25	0.004		0.010
a2			1.65			0.065
a3	0.65		0.85	0.026		0.033
b	0.35		0.48	0.014		0.019
b1	0.19		0.25	0.007		0.010
C	0.25		0.5	0.010		0.020
c1	45° (typ.)					
D	4.8		5.0	0.189		0.197
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		3.81			0.150	
F	3.8		4.0	0.150		0.157
L	0.4		1.27	0.016		0.050
M			0.6			0.024
S	8° (max.)					

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

© The ST logo is a registered trademark of STMicroelectronics

© 2002 STMicroelectronics - Printed in Italy - All Rights Reserved
 STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia
 Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States

© <http://www.st.com>



This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

MC14066B

Quad Analog Switch/Quad Multiplexer

The MC14066B consists of four independent switches capable of controlling either digital or analog signals. This quad bilateral switch is useful in signal gating, chopper, modulator, demodulator and CMOS logic implementation.

The MC14066B is designed to be pin-for-pin compatible with the MC14016B, but has much lower ON resistance. Input voltage swings as large as the full supply voltage can be controlled via each independent control input.

- Triple Diode Protection on All Control Inputs
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Linearized Transfer Characteristics
- Low Noise — $12 \text{ nV}/\sqrt{\text{Cycle}}$, $f \geq 1.0 \text{ kHz}$ typical
- Pin-for-Pin Replacement for CD4016, CD4016, MC14016B
- For Lower R_{ON} , Use The HC4066 High-Speed CMOS Device

MAXIMUM RATINGS (Voltages Referenced to V_{SS}) (Note 2.)

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage Range	-0.5 to +18.0	V
V_{in}, V_{out}	Input or Output Voltage Range (DC or Transient)	-0.5 to $V_{DD} + 0.5$	V
I_{in}	Input Current (DC or Transient) per Control Pin	± 10	mA
I_{SW}	Switch Through Current	± 25	mA
P_D	Power Dissipation, per Package (Note 3.)	500	mW
T_A	Ambient Temperature Range	-55 to +125	°C
T_{stg}	Storage Temperature Range	-65 to +150	°C
T_L	Lead Temperature (8-Second Soldering)	260	°C

2. Maximum Ratings are those values beyond which damage to the device may occur.

3. Temperature Derating:
Plastic "P and D/DW" Packages: - 7.0 mW/°C From 65°C To 125°C

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the range $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$.

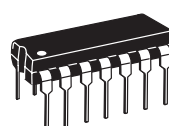
Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}). Unused outputs must be left open.



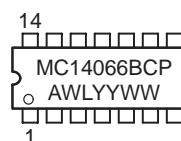
ON Semiconductor

<http://onsemi.com>

MARKING DIAGRAMS



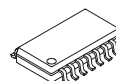
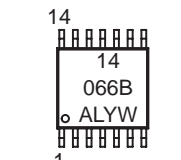
PDIP-14
P SUFFIX
CASE 646



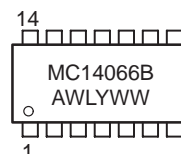
SOIC-14
D SUFFIX
CASE 751A



TSSOP-14
DT SUFFIX
CASE 948G



SOEIAJ-14
F SUFFIX
CASE 965



A = Assembly Location
WL or L = Wafer Lot
YY or Y = Year
WW or W = Work Week

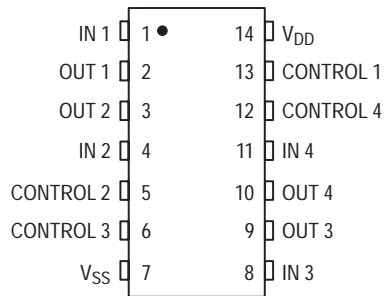
ORDERING INFORMATION

Device	Package	Shipping
MC14066BCP	PDIP-14	2000/Box
MC14066BD	SOIC-14	55/Rail
MC14066BDR2	SOIC-14	2500/Tape & Reel
MC14066BDT	TSSOP-14	96/Rail
MC14066BDTEL	TSSOP-14	2000/Tape & Reel
MC14066BDTR2	TSSOP-14	2500/Tape & Reel
MC14066BF	SOEIAJ-14	See Note 1.
MC14066BFEL	SOEIAJ-14	See Note 1.

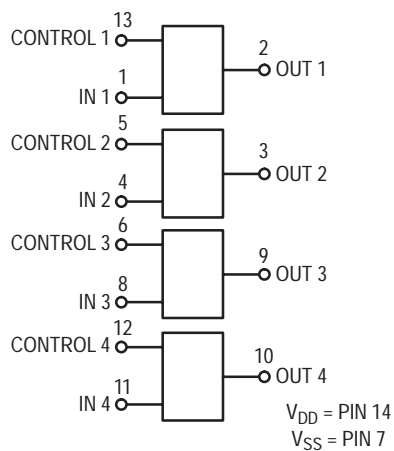
1. For ordering information on the EIAJ version of the SOIC packages, please contact your local ON Semiconductor representative.

MC14066B

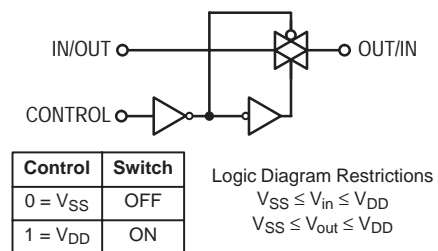
PIN ASSIGNMENT



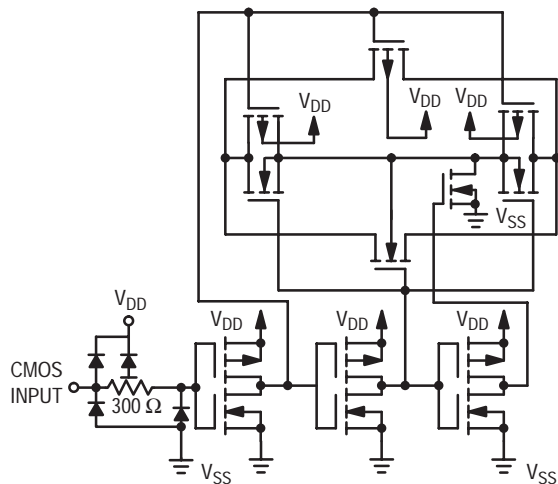
BLOCK DIAGRAM



LOGIC DIAGRAM AND TRUTH TABLE (1/4 OF DEVICE SHOWN)



CIRCUIT SCHEMATIC (1/4 OF CIRCUIT SHOWN)



MC14066B

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	V _{DD}	Test Conditions	- 55°C		25°C			125°C		Unit
				Min	Max	Min	Typ (4.)	Max	Min	Max	

SUPPLY REQUIREMENTS (Voltages Referenced to V_{EE})

Power Supply Voltage Range	V _{DD}	—		3.0	18	3.0	—	18	3.0	18	V
Quiescent Current Per Package	I _{DD}	5.0	Control Inputs: V _{in} = V _{SS} or V _{DD} , Switch I/O: V _{SS} ≤ V _{I/O} ≤ V _{DD} , and ΔV _{switch} ≤ 500 mV (5.)	—	0.25	—	0.005	0.25	—	7.5	μA
		10		—	0.5	—	0.010	0.5	—	15	
		15		—	1.0	—	0.015	1.0	—	30	
Total Supply Current (Dynamic Plus Quiescent, Per Package)	I _{D(AV)}	5.0 10 15	T _A = 25°C only The channel component, (V _{in} - V _{out})/R _{on} , is not included.)	Typical (0.07 μA/kHz) f + I _{DD} (0.20 μA/kHz) f + I _{DD} (0.36 μA/kHz) f + I _{DD}							μA

CONTROL INPUTS (Voltages Referenced to V_{SS})

Low-Level Input Voltage	V _{IL}	5.0	R _{on} = per spec, I _{off} = per spec	—	1.5	—	2.25	1.5	—	1.5	V
		10		—	3.0	—	4.50	3.0	—	3.0	
		15		—	4.0	—	6.75	4.0	—	4.0	
High-Level Input Voltage	V _{IH}	5.0	R _{on} = per spec, I _{off} = per spec	3.5	—	3.5	2.75	—	3.5	—	V
		10		7.0	—	7.0	5.50	—	7.0	—	
		15		11	—	11	8.25	—	11	—	
Input Leakage Current	I _{in}	15	V _{in} = 0 or V _{DD}	—	± 0.1	—	± 0.00001	± 0.1	—	± 1.0	μA
Input Capacitance	C _{in}	—		—	—	—	5.0	7.5	—	—	pF

SWITCHES IN AND OUT (Voltages Referenced to V_{SS})

Recommended Peak-to-Peak Voltage Into or Out of the Switch	V _{I/O}	—	Channel On or Off	0	V _{DD}	0	—	V _{DD}	0	V _{DD}	V _{p-p}
Recommended Static or Dynamic Voltage Across the Switch (5.) (Figure 1)	ΔV _{switch}	—	Channel On	0	600	0	—	600	0	300	mV
Output Offset Voltage	V _{OO}	—	V _{in} = 0 V, No Load	—	—	—	10	—	—	—	μV
ON Resistance	R _{on}	5.0	ΔV _{switch} ≤ 500 mV (5.), V _{in} = V _{IL} or V _{IH} (Control), and V _{in} = 0 to V _{DD} (Switch)	—	800	—	250	1050	—	1200	Ω
		10		—	400	—	120	500	—	520	
		15		—	220	—	80	280	—	300	
ΔON Resistance Between Any Two Channels in the Same Package	ΔR _{on}	5.0		—	70	—	25	70	—	135	Ω
		10		—	50	—	10	50	—	95	
		15		—	45	—	10	45	—	65	
Off-Channel Leakage Current (Figure 6)	I _{off}	15	V _{in} = V _{IL} or V _{IH} (Control) Channel to Channel or Any One Channel	—	± 100	—	± 0.05	± 100	—	± 1000	nA
Capacitance, Switch I/O	C _{I/O}	—	Switch Off	—	—	—	10	15	—	—	pF
Capacitance, Feedthrough (Switch Off)	C _{I/O}	—		—	—	—	0.47	—	—	—	pF

4. Data labeled "Typ" is not to be used for design purposes, but is intended as an indication of the IC's potential performance.

5. For voltage drops across the switch (ΔV_{switch}) > 600 mV (> 300 mV at high temperature), excessive V_{DD} current may be drawn; i.e. the current out of the switch may contain both V_{DD} and switch input components. The reliability of the device will be unaffected unless the Maximum Ratings are exceeded. (See first page of this data sheet.)

MC14066B

ELECTRICAL CHARACTERISTICS ^(6.) ($C_L = 50 \text{ pF}$, $T_A = 25^\circ\text{C}$ unless otherwise noted.)

Characteristic	Symbol	V_{DD} Vdc	Min	Typ ^(7.)	Max	Unit
Propagation Delay Times Input to Output ($R_L = 10 \text{ k}\Omega$) $V_{SS} = 0 \text{ Vdc}$ $t_{PLH}, t_{PHL} = (0.17 \text{ ns/pF}) C_L + 15.5 \text{ ns}$ $t_{PLH}, t_{PHL} = (0.08 \text{ ns/pF}) C_L + 6.0 \text{ ns}$ $t_{PLH}, t_{PHL} = (0.06 \text{ ns/pF}) C_L + 4.0 \text{ ns}$	t_{PLH}, t_{PHL}	5.0 10 15	— — —	20 10 7.0	40 20 15	ns
Control to Output ($R_L = 1 \text{ k}\Omega$) (Figure 2) Output "1" to High Impedance	t_{PHZ}	5.0 10 15	— — —	40 35 30	80 70 60	ns
Output "0" to High Impedance	t_{PLZ}	5.0 10 15	— — —	40 35 30	80 70 60	ns
High Impedance to Output "1"	t_{PZH}	5.0 10 15	— — —	60 20 15	120 40 30	ns
High Impedance to Output "0"	t_{PZL}	5.0 10 15	— — —	60 20 15	120 40 30	ns
Second Harmonic Distortion $V_{SS} = -5 \text{ Vdc}$ ($V_{in} = 1.77 \text{ Vdc}$, RMS Centered @ 0.0 Vdc , $R_L = 10 \text{ k}\Omega$, $f = 1.0 \text{ kHz}$)	—	5.0	—	0.1	—	%
Bandwidth (Switch ON) (Figure 3) $V_{SS} = -5 \text{ Vdc}$ ($R_L = 1 \text{ k}\Omega$, $20 \text{ Log } (V_{out}/V_{in}) = -3 \text{ dB}$, $C_L = 50 \text{ pF}$, $V_{in} = 5 \text{ V}_{p-p}$)	—	5.0	—	65	—	MHz
Feedthrough Attenuation (Switch OFF) $V_{SS} = -5 \text{ Vdc}$ ($V_{in} = 5 \text{ V}_{p-p}$, $R_L = 1 \text{ k}\Omega$, $f_{in} = 1.0 \text{ MHz}$) (Figure 3)	—	5.0	—	-50	—	dB
Channel Separation (Figure 4) $V_{SS} = -5 \text{ Vdc}$ ($V_{in} = 5 \text{ V}_{p-p}$, $R_L = 1 \text{ k}\Omega$, $f_{in} = 8.0 \text{ MHz}$) (Switch A ON, Switch B OFF)	—	5.0	—	-50	—	dB
Crosstalk, Control Input to Signal Output (Figure 5) $V_{SS} = -5 \text{ Vdc}$ ($R_1 = 1 \text{ k}\Omega$, $R_L = 10 \text{ k}\Omega$, Control $t_{TLH} = t_{THL} = 20 \text{ ns}$)	—	5.0	—	300	—	mV_{p-p}

6. The formulas given are for the typical characteristics only at 25°C .

7. Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance.

TEST CIRCUITS

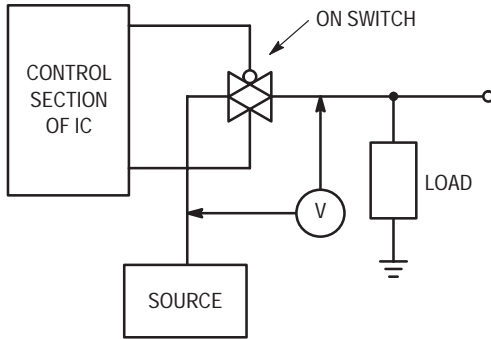


Figure 1. ΔV Across Switch

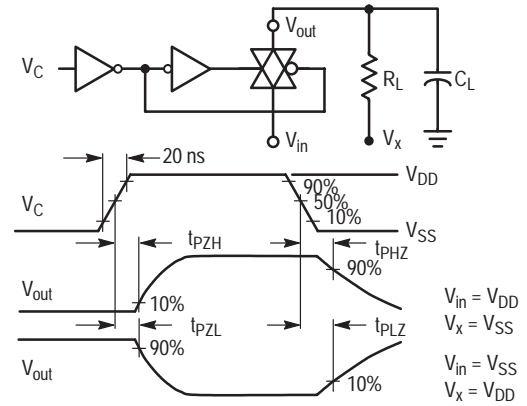


Figure 2. Turn-On Delay Time Test Circuit and Waveforms

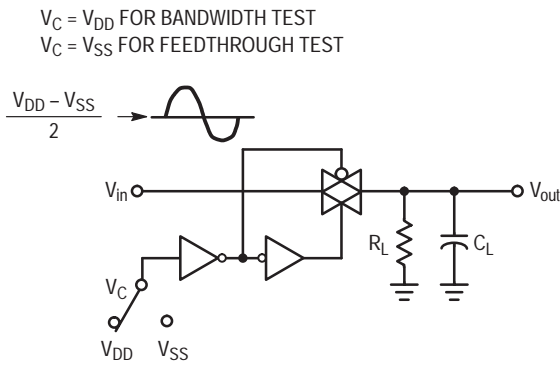


Figure 3. Bandwidth and Feedthrough Attenuation

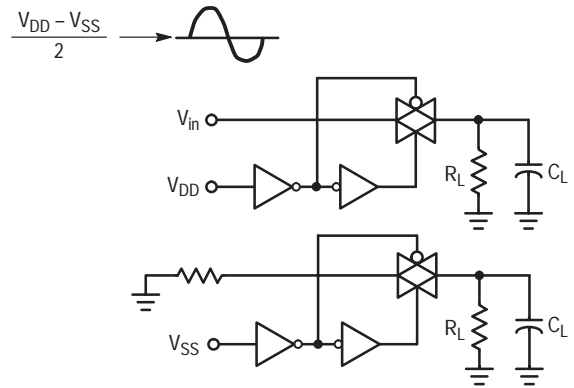


Figure 4. Channel Separation

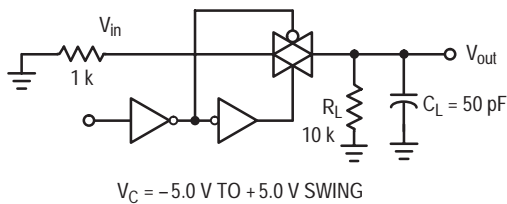


Figure 5. Crosstalk, Control to Output

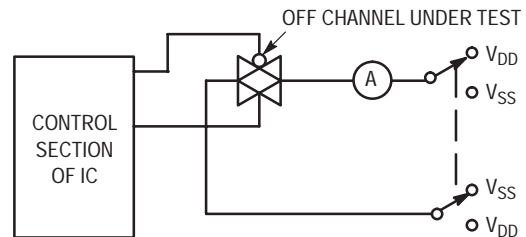


Figure 6. Off Channel Leakage

MC14066B

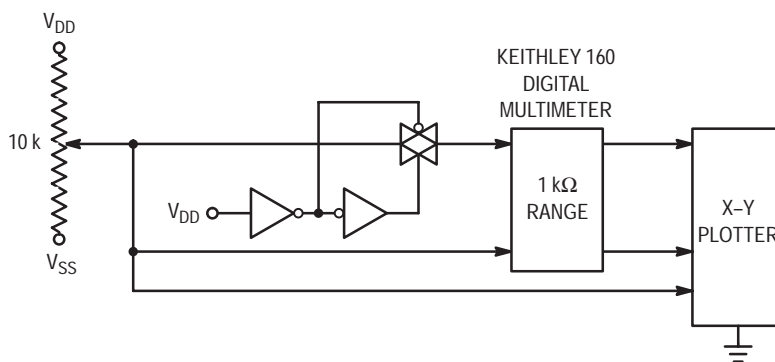


Figure 7. Channel Resistance (R_{ON}) Test Circuit

TYPICAL RESISTANCE CHARACTERISTICS

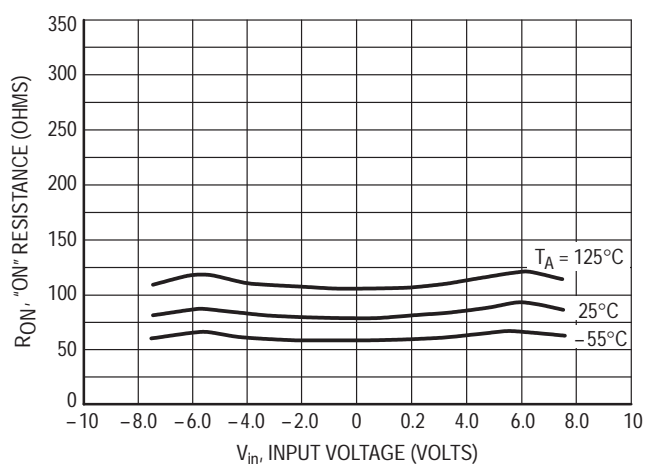


Figure 8. $V_{DD} = 7.5\text{ V}$, $V_{SS} = -7.5\text{ V}$

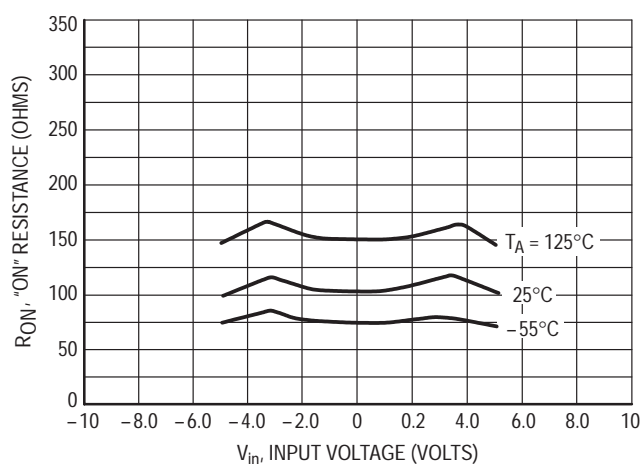


Figure 9. $V_{DD} = 5.0\text{ V}$, $V_{SS} = -5.0\text{ V}$

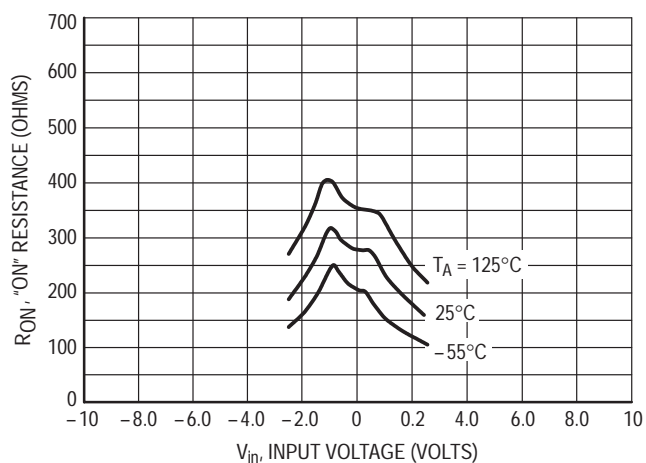


Figure 10. $V_{DD} = 2.5\text{ V}$, $V_{SS} = -2.5\text{ V}$

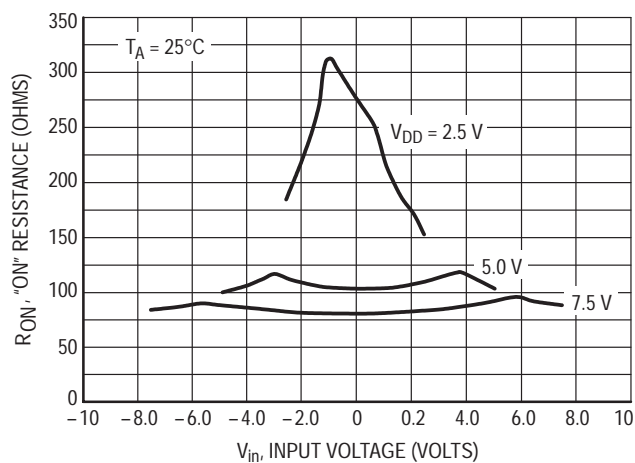


Figure 11. Comparison at 25°C, $V_{DD} = -V_{SS}$

APPLICATIONS INFORMATION

Figure A illustrates use of the Analog Switch. The 0–to–5 volt digital control signal is used to directly control a 5 volt peak-to-peak analog signal.

The digital control logic levels are determined by V_{DD} and V_{SS} . The V_{DD} voltage is the logic high voltage, the V_{SS} voltage is logic low. For the example, $V_{DD} = +5\text{ V} = \text{logic high}$ at the control inputs; $V_{SS} = \text{GND} = 0\text{ V} = \text{logic low}$.

The maximum analog signal level is determined by V_{DD} and V_{SS} . The analog voltage must not swing higher than V_{DD} or lower than V_{SS} .

The example shows a 5 volt peak-to-peak signal which allows no margin at either peak. If voltage transients above

V_{DD} and/or below V_{SS} are anticipated on the analog channels, external diodes (D_X) are recommended as shown in Figure B. These diodes should be small signal types able to absorb the maximum anticipated current surges during clipping.

The *absolute* maximum potential difference between V_{DD} and V_{SS} is 18.0 volts. Most parameters are specified up to 15 volts which is the *recommended* maximum difference between V_{DD} and V_{SS} .

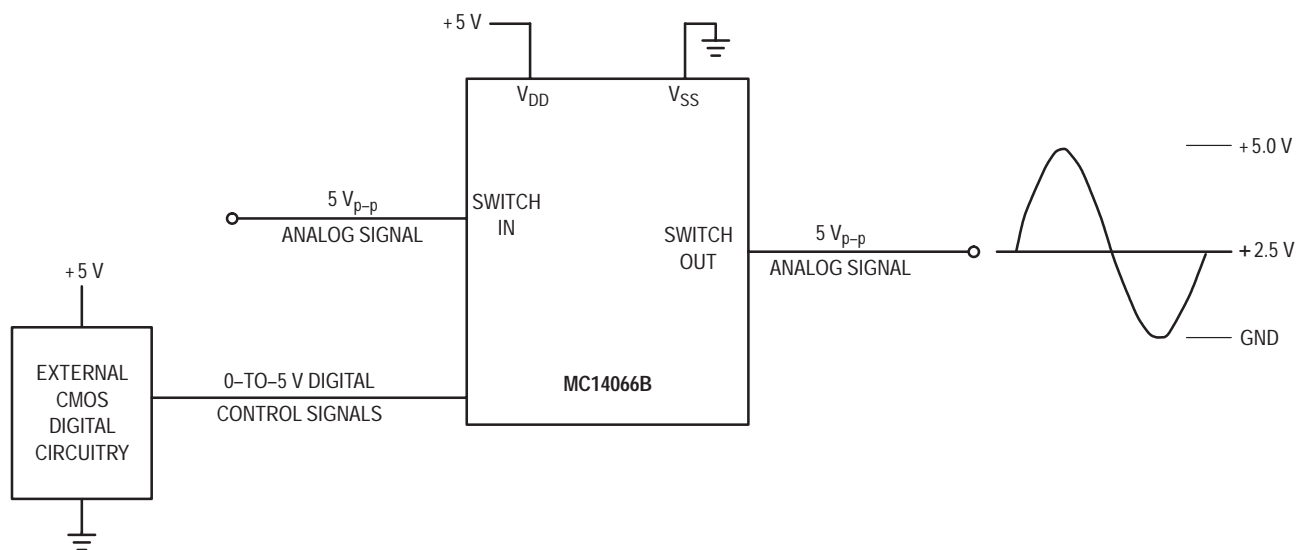


Figure A. Application Example

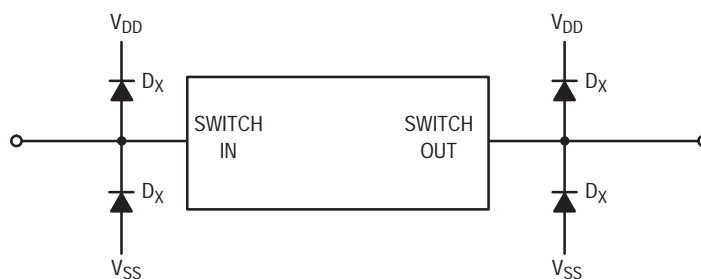
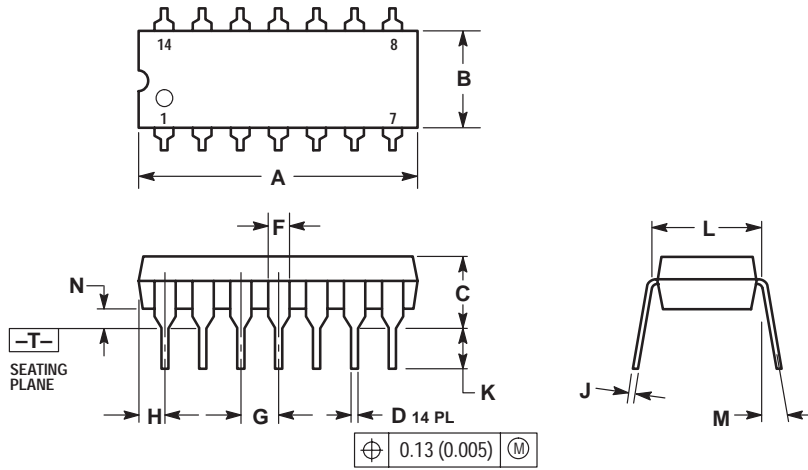


Figure B. External Germanium or Schottky Clipping Diodes

MC14066B

PACKAGE DIMENSIONS

P SUFFIX
PLASTIC DIP PACKAGE
CASE 646-06
ISSUE M



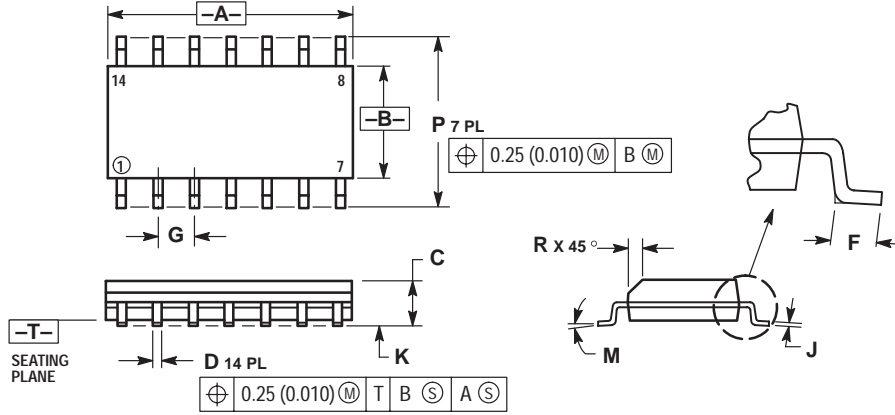
- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
 2. CONTROLLING DIMENSION: INCH.
 3. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
 4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
 5. ROUNDED CORNERS OPTIONAL.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.715	0.770	18.16	18.80
B	0.240	0.260	6.10	6.60
C	0.145	0.185	3.69	4.69
D	0.015	0.021	0.38	0.53
F	0.040	0.070	1.02	1.78
G	0.100 BSC		2.54 BSC	
H	0.052	0.095	1.32	2.41
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.290	0.310	7.37	7.87
M	---	10°	---	10°
N	0.015	0.039	0.38	1.01

MC14066B

PACKAGE DIMENSIONS

D SUFFIX PLASTIC SOIC PACKAGE CASE 751A-03 ISSUE F



NOTES:

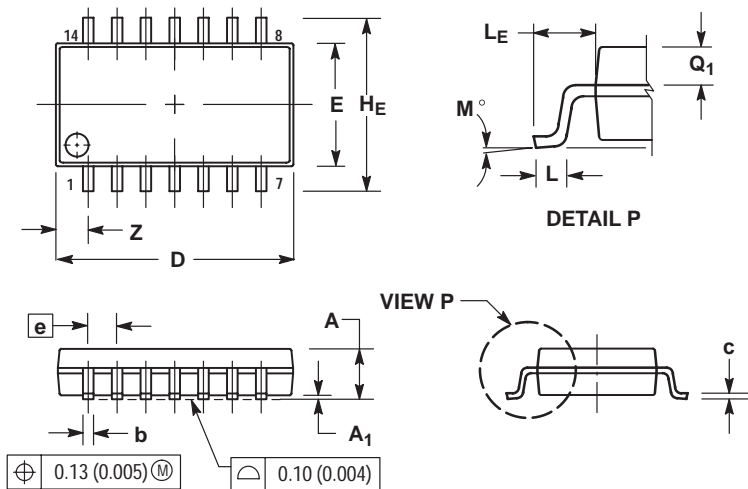
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 (0.005) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	8.55	8.75	0.337	0.344
B	3.80	4.00	0.150	0.157
C	1.35	1.75	0.054	0.068
D	0.35	0.49	0.014	0.019
F	0.40	1.25	0.016	0.049
G	1.27 BSC		0.050 BSC	
J	0.19	0.25	0.008	0.009
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	5.80	6.20	0.228	0.244
R	0.25	0.50	0.010	0.019

MC14066B

PACKAGE DIMENSIONS


F SUFFIX PLASTIC EIAJ SOIC PACKAGE CASE 965-01 ISSUE O



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS D AND E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS AND ARE MEASURED AT THE PARTING LINE. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.15 (0.006) PER SIDE.
4. TERMINAL NUMBERS ARE SHOWN FOR REFERENCE ONLY.
5. THE LEAD WIDTH DIMENSION (b) DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE LEAD WIDTH DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSIONS AND ADJACENT LEAD TO BE 0.46 (0.018).

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	—	2.05	—	0.081
A ₁	0.05	0.20	0.002	0.008
b	0.35	0.50	0.014	0.020
c	0.18	0.27	0.007	0.011
D	9.90	10.50	0.390	0.413
E	5.10	5.45	0.201	0.215
e	1.27 BSC		0.050 BSC	
H _E	7.40	8.20	0.291	0.323
0.50	0.50	0.85	0.020	0.033
L _F	1.10	1.50	0.043	0.059
M	0°	10°	0°	10°
Q ₁	0.70	0.90	0.028	0.035
Z	—	1.42	—	0.056

ON Semiconductor and  are trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer.

PUBLICATION ORDERING INFORMATION

NORTH AMERICA Literature Fulfillment:

Literature Distribution Center for ON Semiconductor
P.O. Box 5163, Denver, Colorado 80217 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: ONlit@hibbertco.com
Fax Response Line: 303-675-2167 or 800-344-3810 Toll Free USA/Canada

N. American Technical Support: 800-282-9855 Toll Free USA/Canada

EUROPE: LDC for ON Semiconductor – European Support

German Phone: (+1) 303-308-7140 (M–F 1:00pm to 5:00pm Munich Time)
Email: ONlit-german@hibbertco.com
French Phone: (+1) 303-308-7141 (M–F 1:00pm to 5:00pm Toulouse Time)
Email: ONlit-french@hibbertco.com
English Phone: (+1) 303-308-7142 (M–F 12:00pm to 5:00pm UK Time)
Email: ONlit@hibbertco.com

EUROPEAN TOLL-FREE ACCESS*: 00-800-4422-3781

*Available from Germany, France, Italy, England, Ireland

CENTRAL/SOUTH AMERICA:

Spanish Phone: 303-308-7143 (Mon–Fri 8:00am to 5:00pm MST)
Email: ONlit-spanish@hibbertco.com

ASIA/PACIFIC: LDC for ON Semiconductor – Asia Support

Phone: 303-675-2121 (Tue–Fri 9:00am to 1:00pm, Hong Kong Time)
Toll Free from Hong Kong & Singapore:
001-800-4422-3781
Email: ONlit-asia@hibbertco.com

JAPAN: ON Semiconductor, Japan Customer Focus Center
4-32-1 Nishi-Gotanda, Shinagawa-ku, Tokyo, Japan 141-8549
Phone: 81-3-5740-2745
Email: r14525@onsemi.com

ON Semiconductor Website: <http://onsemi.com>

For additional information, please contact your local Sales Representative.

This datasheet has been downloaded from:

www.DatasheetCatalog.com

Datasheets for electronic components.

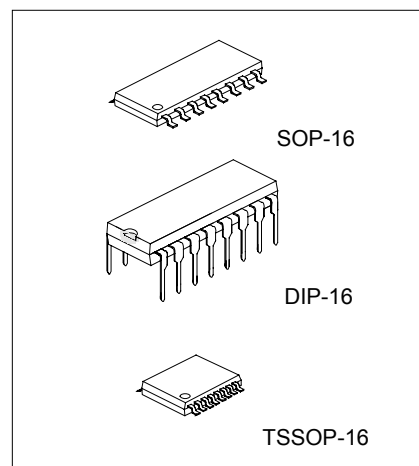
ANALOG MULTIPLEXERS/ DEMULTIPLEXERS

DESCRIPTION

The UTC **4053** are Triple SPDT analog multiplexers for application as digitally-controlled analog switches.

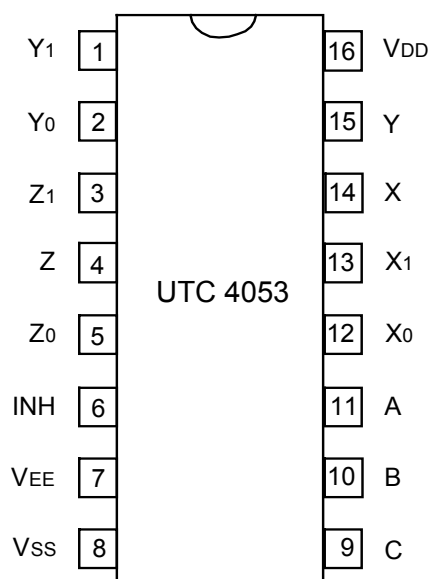
FEATURES

- * Analog Voltage Range ($V_{DD} - V_{EE}$) = 3.0 ~ 18 V
Note: V_{EE} must be $\leq V_{SS}$
- * Linearized Transfer Characteristics
- * Pin-to-Pin Replacement for CD4053

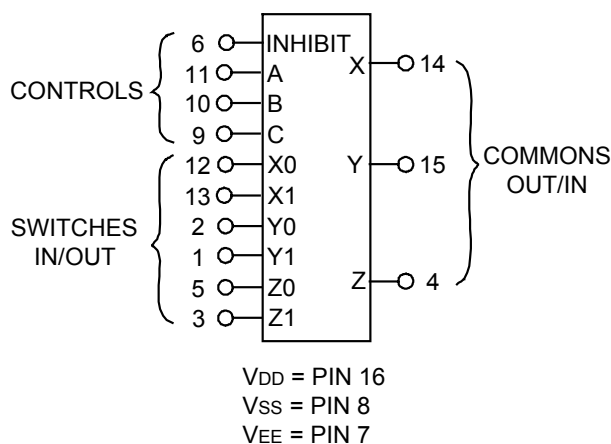


*Pb-free plating product number: 4053L

PIN CONFIGURATIONS



UTC 4053 Triple 2-Channel Analog Multiplexer/Demultiplexer



Note: Control Inputs referenced to V_{SS} , Analog Inputs and Outputs reference to V_{EE} . V_{EE} must be $\leq V_{SS}$.

ABSOLUTE MAXIMUM RATINGS*

PARAMETER	SYMBOL	RATINGS	UNIT
DC Supply Voltage (Referenced to V_{EE} , $V_{SS} \geq V_{EE}$)	V_{DD}	-0.5 ~ +18.0	V
Input or Output Voltage (DC or Transient) (Referenced to V_{SS} for Control Inputs and V_{EE} for Switch I/O)	V_{in} , V_{out}	-0.5 ~ $V_{DD} + 0.5$	V
Input Current (DC or Transient), per Control Pin	I_{in}	± 10	mA
Switch Through Current	I_{SW}	± 25	mA
Power Dissipation. Per Package**	P_D	500	mW
Storage Temperature	T_{stg}	-65 ~ +150	$^{\circ}\text{C}$
Lead Temperature (8 - Second Soldering)	T_{Lead}	260	$^{\circ}\text{C}$

* Maximum Ratings are those values beyond which damage to the device may occur.

** Temperature Derating: "DIP and SOP" Packages: - 7.0 mW/ $^{\circ}\text{C}$ From 65 $^{\circ}\text{C}$ ~ 125 $^{\circ}\text{C}$

ELECTRICAL CHARACTERISTICS

($T_a = 25^{\circ}\text{C}$, unless otherwise indicated.)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP#	MAX	UNIT
SUPPLY REQUIREMENTS (Voltages Referenced to V_{EE})						
Power Supply Voltage Range	V_{DD}	$V_{DD} - 3.0 \geq V_{SS} \geq V_{EE}$	3.0		18	V
Quiescent Current per Package	I_{DD}	Control Inputs: $V_{in} = V_{SS}$ or V_{DD} Switch I/O: $V_{EE} \leq V_{IO} \leq V_{DD}$, and $\Delta V_{switch} \leq 500\text{mV}^*$ $V_{DD} = 5.0\text{V}$ $V_{DD} = 10\text{V}$ $V_{DD} = 15\text{V}$		0.005 0.010 0.015	5.0 10 20	μA
Total Supply Current (Dynamic Plus Quiescent, Per Package)	$I_{D(AV)}$	$T_a = 25^{\circ}\text{C}$ only (The channel component, $(V_{in} - V_{out})/R_{on}$, is not included.) $V_{DD} = 5.0\text{V}$ $V_{DD} = 10\text{V}$ $V_{DD} = 15\text{V}$		(0.07 $\mu\text{A}/\text{kHz}$) $f + I_{DD}$ Typical (0.20 $\mu\text{A}/\text{kHz}$) $f + I_{DD}$ (0.36 $\mu\text{A}/\text{kHz}$) $f + I_{DD}$		μA

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP#	MAX	UNIT
CONTROL INPUTS – INHIBIT A, B, C (Voltages Referenced to V_{SS})						
Low – Level Input Voltage	V _{IL}	Ron= per spec, Ioff = per spec V _{DD} =5.0V V _{DD} =10V V _{DD} =15V		2.25 4.50 6.75	1.5 3.0 4.0	V
High – Level Input Voltage	V _{IH}	Ron= per spec, Ioff = per spec V _{DD} =5.0V V _{DD} =10V V _{DD} =15V	3.5 7.0 11	2.75 5.50 8.25		V
Input Leakage Current	I _{in}	V _{in} = 0 or V _{DD} , V _{DD} =15V		±0.00001	±0.1	μA
Input Capacitance	C _{in}			5.0	7.5	pF
SWITCHES IN/OUT AND COMMONS OUT/IN -- X, Y, Z (Voltages Referenced to V_{EE})						
Recommended Peak-to-Peak Voltage Into or Out of the Switch	V _{I/O}	Channel On or Off	0		V _{DD}	V _{PP}
Recommended Static or Dynamic Voltage Across the Switch** (Figure 3)	ΔV _{switch}	Channel On	0		600	mV
Output Offset Voltage	V _{OO}	V _{in} = 0V, No Load		10		μV
ON Resistance	Ron	ΔV _{switch} ≤ 500mV* V _{in} = V _{IL} or V _{IH} (Control), and V _{in} = 0 to V _{DD} (Switch) V _{DD} =5.0V V _{DD} =10V V _{DD} =15V		250 120 80	1050 500 280	Ω
Δ ON Resistance Between Any Two Channels in the Same Package	ΔRon	V _{DD} =5.0V V _{DD} =10V V _{DD} =15V		25 10 10	70 50 45	Ω
Off-Channel Leakage Current (Figure 8)	Ioff	V _{in} = V _{IL} or V _{IH} (Control) Channel to Channel or Any One Channel, V _{DD} =15V		±0.05	±100	nA
Capacitance, Switch I/O	C _{I/O}	Inhibit = V _{DD}		10		pF
Capacitance, Common O/I	C _{O/I}	Inhibit = V _{DD}		17		pF
Capacitance, Feedthrough (Channel Off)	C _{I/O}	Pins Not Adjacent Pins Adjacent		0.15 0.47		pF

#Data labeled "Typ" is not to be used for design purposes, but is intended as an indication of the IC's potential performance.

* For voltage drops across the switch (ΔV_{switch}) > 600 mV (> 300 mV at high temperature), excessive V_{DD} current may be drawn, i.e. the current out of the switch may contain both V_{DD} and switch input components. The reliability of the device will be unaffected unless the Maximum Ratings are exceeded. (See second page of this data sheet.)

ELECTRICAL CHARACTERISTICS*(C_L = 50pF, T_a=25°C, V_{EE} ≤ V_{SS}, unless otherwise indicated.)

PARAMETER	SYMBOL	V _{DD} - V _{EE} Vdc	TEST CONDITIONS	MIN	TYP#	MAX	UNIT
Propagation Delay Times (Figure 4) Switch Input to Switch Output (R _L = 10 kΩ)	t _{PLH} , t _{PHL}	5.0 10 15	t _{PLH} , t _{PHL} = (0.17 ns/pF) C _L + 16.5 ns t _{PLH} , t _{PHL} = (0.08 ns/pF) C _L + 4.0 ns t _{PLH} , t _{PHL} = (0.06 ns/pF) C _L + 3.0 ns		25 8.0 6.0	65 20 15	ns
Inhibit to Output	t _{PHZ} , t _{PLZ} t _{PZH} , t _{PZL}	5.0 10 15	(R _L =10kΩ, V _{EE} =V _{SS})Output "1" or "0" to High Impedance, or High Impedance to "1" or "0" Level		275 140 110	550 280 220	ns
Control Input to Output	t _{PLH} , t _{PHL}	5.0 10 15	R _L = 10 kΩ, V _{EE} = V _{SS}		300 120 80	600 240 160	ns
Second Harmonic Distortion		10	R _L = 10KΩ, f = 1 kHz, Vin = 5 V _{PP}		0.07		%
Bandwidth (Figure 5)	BW	10	R _L = 1kΩ, Vin = 1/2 (V _{DD} -V _{EE}) p-p, C _L = 50pF, 20 Log (Vout/Vin) = -3dB		17		MHz
Off Channel Feedthrough Attenuation (Figure 5)		10	R _L = 1KΩ, Vin = 1/2 (V _{DD} - V _{EE}) p-p fin = 55 MHz		-50		dB
Channel Separation (Figure 6)		10	R _L = 1 kΩ, Vin = 1/2 (V _{DD} -V _{EE}) p-p fin = 3.0 MHz		-50		dB
Crosstalk, Control Input to Common O/I (Figure 7)		10	R ₁ = 1 kΩ, R _L = 10 kΩ Control t _{TLH} = t _{THL} = 20 ns, Inhibit = V _{SS})		75		mV

* The formulas given are for the typical characteristics only at 25°C.

Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance.

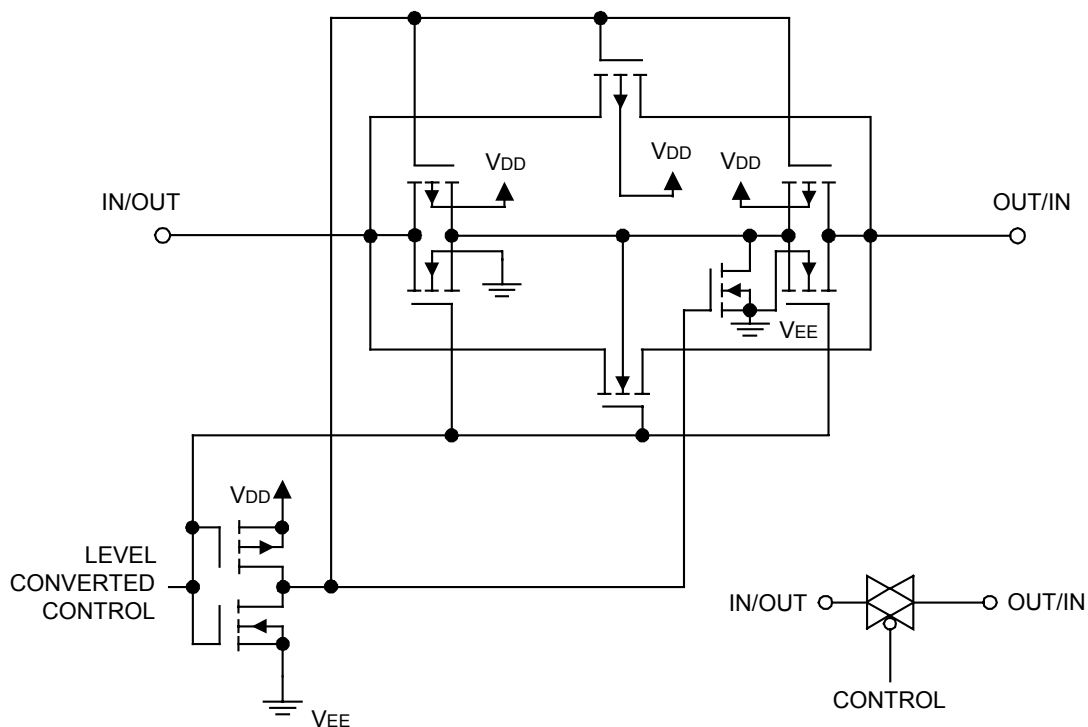


Figure 1. Switch Circuit Schematic

TRUTH TABLE

Control Inputs				ON Switches		
Inhibit	Select					
	C	B	A	UTC 4053		
0	0	0	0	Z0	Y0	X0
0	0	0	1	Z0	Y0	X1
0	0	1	0	Z0	Y1	X0
0	0	1	1	Z0	Y1	X1
0	1	0	0	Z1	Y0	X0
0	1	0	1	Z1	Y0	X1
0	1	1	0	Z1	Y1	X0
0	1	1	1	Z1	Y1	X1
1	x	x	x	None		

x = Don't Care

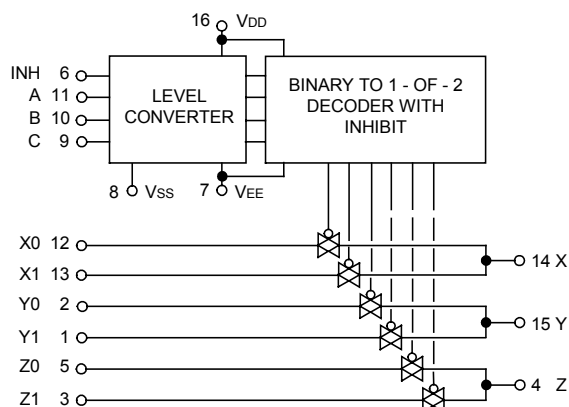
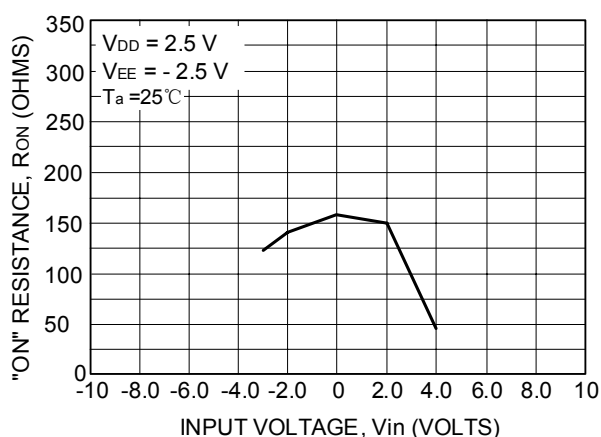
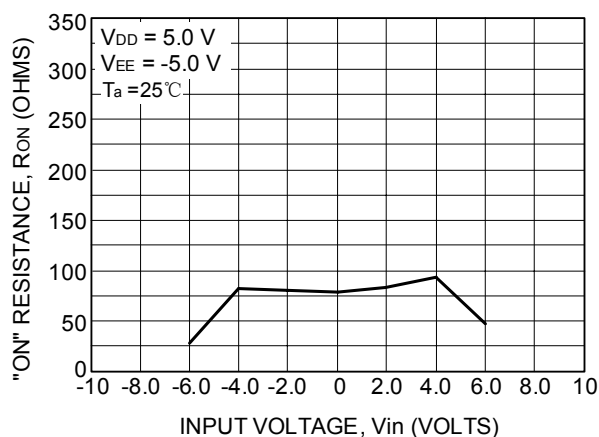
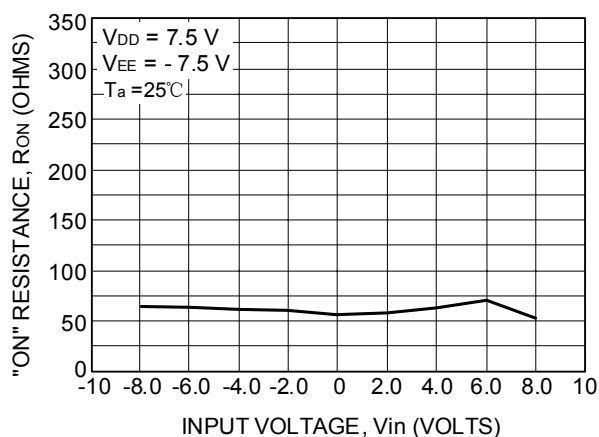


Figure 2. UTC 4053 Functional Diagram



UTC assumes no responsibility for equipment failures that result from using products at values that exceed, even momentarily, rated values (such as maximum ratings, operating condition ranges, or other parameters) listed in products specifications of any and all UTC products described or contained herein. UTC products are not designed for use in life support appliances, devices or systems where malfunction of these products can be reasonably expected to result in personal injury. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice.

This datasheet has been downloaded from:

www.DatasheetCatalog.com

Datasheets for electronic components.

DATA SHEET

For a complete data sheet, please also download:

- The IC06 74HC/HCT/HCU/HCMOS Logic Family Specifications
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Information
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Outlines

74HC/HCT04 Hex inverter

Product specification
File under Integrated Circuits, IC06

September 1993

Hex inverter

74HC/HCT04

FEATURES

- Output capability: standard
- I_{CC} category: SSI

GENERAL DESCRIPTION

The 74HC/HCT04 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A. The 74HC/HCT04 provide six inverting buffers.

QUICK REFERENCE DATA

GND = 0 V; $T_{amb} = 25\text{ }^{\circ}\text{C}$; $t_r = t_f = 6\text{ ns}$

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PHL}/t_{PLH}	propagation delay nA to nY	$C_L = 15\text{ pF}$; $V_{CC} = 5\text{ V}$	7	8	ns
C_I	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per gate	notes 1 and 2	21	24	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

$\sum (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is $V_I = \text{GND to } V_{CC}$
For HCT the condition is $V_I = \text{GND to } V_{CC} - 1.5\text{ V}$

ORDERING INFORMATION

See *"74HC/HCT/HCU/HCMOS Logic Package Information"*.

Hex inverter

74HC/HCT04

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 3, 5, 9, 11, 13	1A to 6A	data inputs
2, 4, 6, 8, 10, 12	1Y to 6Y	data outputs
7	GND	ground (0 V)
14	V _{CC}	positive supply voltage

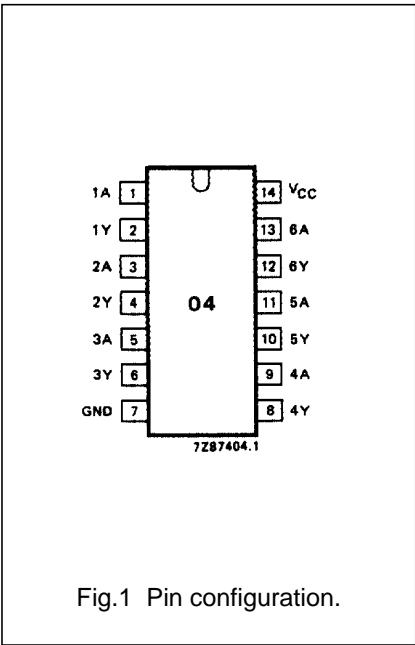


Fig.1 Pin configuration.

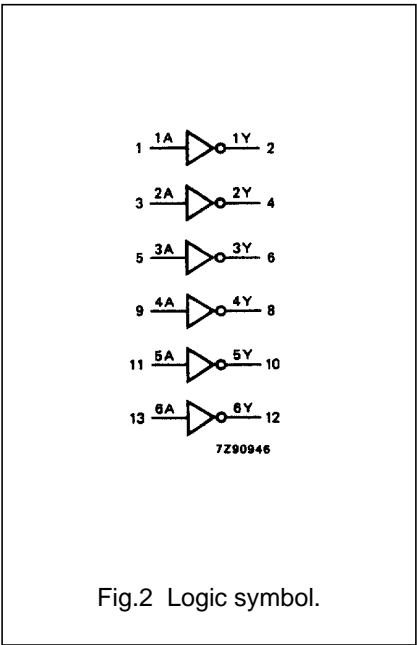


Fig.2 Logic symbol.

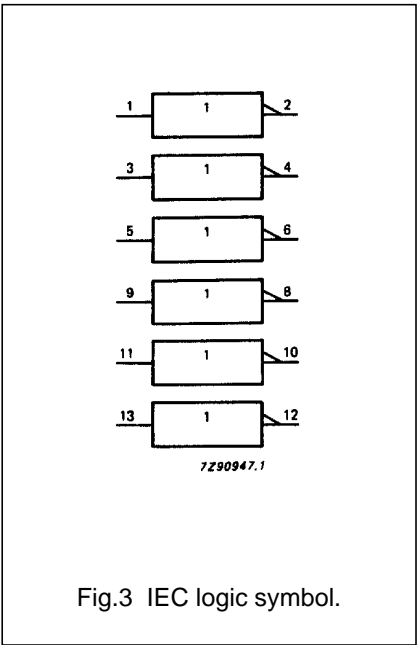


Fig.3 IEC logic symbol.

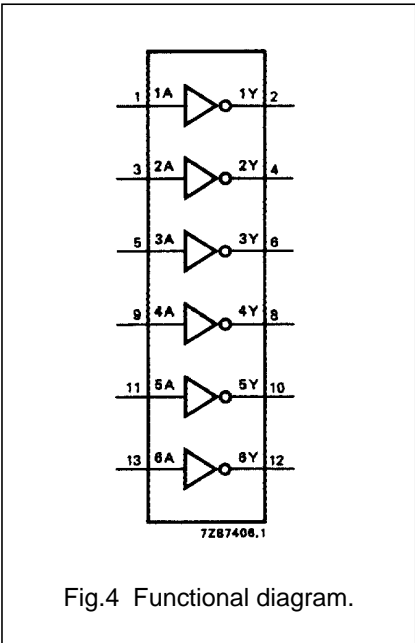


Fig.4 Functional diagram.

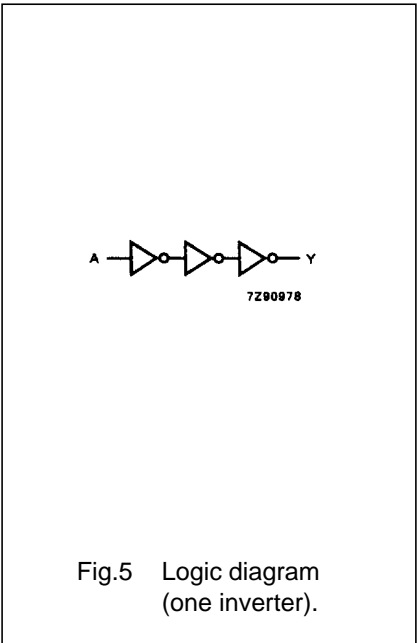


Fig.5 Logic diagram (one inverter).

FUNCTION TABLE

INPUT	OUTPUT
nA	nY
L	H
H	L

Notes

- 1. H = HIGH voltage level
- L = LOW voltage level

Hex inverter

74HC/HCT04

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see *"74HC/HCT/HCU/HCMOS Logic Family Specifications"*.

Output capability: standard

I_{CC} category: SSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)							UNIT	TEST CONDITIONS	
		74HC								V _{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125				
		min.	typ.	max.	min.	max.	min.	max.			
t _{PHL} / t _{PLH}	propagation delay nA to nY		25 9 7	85 17 14		105 21 18		130 26 22	ns	2.0 4.5 6.0	Fig.6
t _{THL} / t _{TLH}	output transition time		19 7 6	75 15 13		95 19 16		110 22 19	ns	2.0 4.5 6.0	Fig.6

Hex inverter

74HC/HCT04

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see *“74HC/HCT/HCU/HCMOS Logic Family Specifications”*.

Output capability: standard
I_{CC} category: SSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications.
To determine ΔI_{CC} per unit, multiply this value by the unit load coefficient shown in the table below.

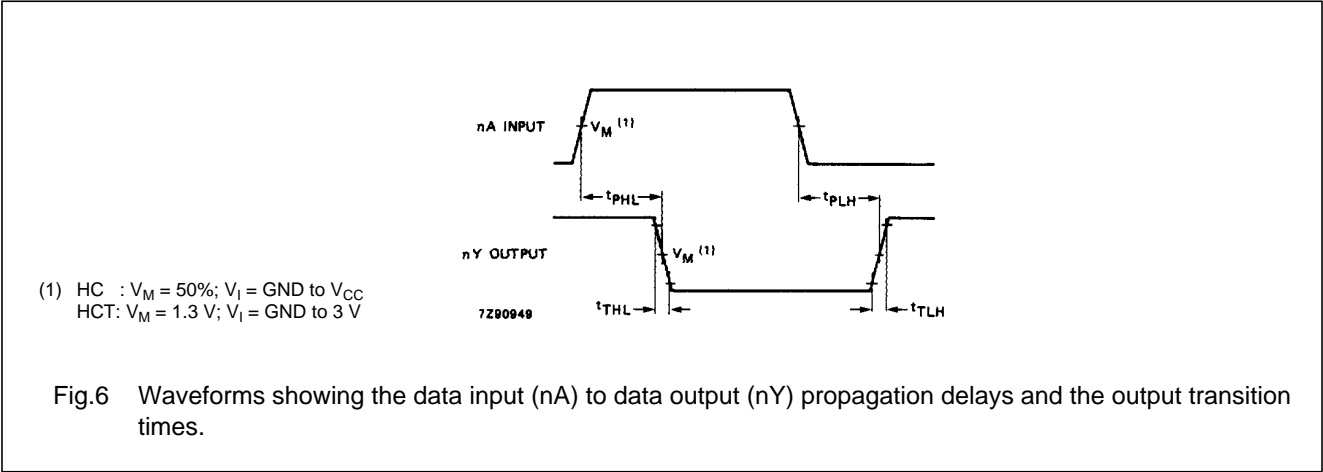
INPUT	UNIT LOAD COEFFICIENT
nA	1.20

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)							UNIT	TEST CONDITIONS	
		74HCT								V _{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125				
		min.	typ.	max.	min.	max.	min.	max.			
t _{PHL} / t _{PLH}	propagation delay nA to nY		10	19		24		29	ns	4.5	Fig.6
t _{THL} / t _{TLH}	output transition time		7	15		19		22	ns	4.5	Fig.6

AC WAVEFORMS



PACKAGE OUTLINES

See *“74HC/HCT/HCU/HCMOS Logic Package Outlines”*.

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.